

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
1 February 2001 (01.02.2001)

PCT

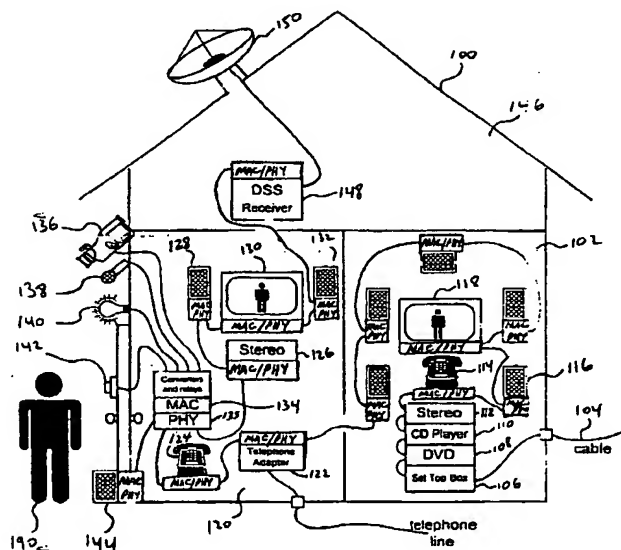
(10) International Publication Number  
WO 01/08366 A1

- (51) International Patent Classification<sup>7</sup>: H04L 25/49 (74) Agents: DE GUZMAN, Arnold, M. et al.; Fenwick & West LLP, Two Palo Alto Square, Palo Alto, CA 94306 (US).
- (21) International Application Number: PCT/US00/19979
- (22) International Filing Date: 21 July 2000 (21.07.2000) (81) Designated States (national): JP, KR.
- (25) Filing Language: English (84) Designated States (regional): European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).
- (26) Publication Language: English
- (30) Priority Data: 60/145,461 23 July 1999 (23.07.1999) US Published:  
— With international search report.  
— Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.
- (71) Applicant: CENTILLIUM COMMUNICATIONS, INC. [US/US]; 47211 Lakeview Boulevard, Fremont, CA 94538 (US).
- (72) Inventors: HOOVER, D., Robert; -. ZDYBEL, Frank, Jr.; -. LYNN, Kerry, E.; -. BHATNAGAR, Manoj; -. BAUDOT, Marie-Dominique; -. For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: APPARATUS AND METHOD FOR MEDIA ACCESS CONTROL



WO 01/08366 A1



(57) Abstract: A protocol and architecture for a synchronous logical ring network that operates on existing physical twisted-pair telephone topologies. In a first embodiment, the invention provides a method and apparatus to communicate using symbols generated by a source device on a network. In a second embodiment, the invention provides a method and apparatus to transmit a command stream generated by a source device in a network. In a third embodiment, the invention provides a method and apparatus to transmit an audio stream generated by a source device on a network. In a fourth embodiment, the invention provides a method and apparatus to transmit an asynchronous packet stream generated by a source device on a network. In a fifth embodiment, the invention provides a method and apparatus to transmit a telephone stream generated by a source device on a network for reception in the network. In a sixth embodiment, the invention provides a method and apparatus to determine a clock offset on a logical ring network. In a seventh embodiment, the invention provides a method and apparatus to interface stream information between one or more network control protocols and a network physical layer. In an eighth

embodiment, the invention provides a method and apparatus to elect a common network clock device on a logical ring network. In a ninth embodiment, the invention provides a method and apparatus to allocate a set of lanes in a frame containing a plurality of lanes, in a network connecting a plurality of devices. In a tenth embodiment, the invention provides a method and apparatus to transmit bi-directional synchronous data streams on a time-division multiplexed access (TDMA)-oriented network connecting a plurality of devices. In an eleventh embodiment, the invention provides a method and apparatus for broadcasting device identification during startup of a device in a network connecting a plurality of devices. In a twelfth embodiment, the invention provides a method and apparatus for structuring the data architecture of a device read only memory (ROM) in a network connecting a plurality of devices.

BEST AVAILABLE COPY

## Apparatus and Method for Media Access Control

5

10

### CROSS-REFERENCE TO RELATED APPLICATIONS

This application relates to and claims priority from co-pending U. S. Provisional Patent Application Serial No. 60/145,461, entitled "Media Access Controller," filed on July 23, 1999, which is hereby incorporated by reference. This application also relates to co-pending U. S. Patent Application Serial No. 09/079,914, entitled "Synchronous Network for Digital Media Streams," filed on May 15, 1998, which claimed priority from U. S. Provisional Patent Application Serial No. 60/050,933, entitled "Synchronous Network for Digital Media Streams," which are hereby incorporated by reference.

20

### BACKGROUND OF THE INVENTION

#### 1. Field of the Invention

This invention relates generally to network interconnection and control of multiple electronic and consumer devices, and in particular to synchronous networks optimized for transmission of digital media streams.

25

#### 2. Nature of the Problem

##### **Interconnecting Devices for Distribution of Audio, Video, other Media/Data, and Control Information**

30

Practical "smart home" networking of consumer electronics devices has not yet arrived. One of the fundamental obstacles to such networking is the difficulty and expense of

interconnecting various types of consumer electronics devices within a single room, or throughout an entire home or other environment, such as a hotel, apartment building, car, boat, or recreational vehicle. Moreover, when such devices are interconnected, there remain problems in distributing control information, audio, video, and other types of media/data, in a sufficiently flexible manner. These problems do not permit users to switch sources, record programs, schedule events, and perform other advanced tasks with the ease of use provided by a simple television and remote control.

For example, existing home theater and home automation systems interconnect devices with dedicated media-specific cables (audio and video cables, speaker wire, etc.), and provide separate controls (e.g., infrared remote controls, X10 control networks, and so forth). These home theater systems quickly become unwieldy as devices are added (including the "spaghetti" of interconnecting cables), in part because such devices typically do not all "speak the same language." Users are faced with the complex task of having to "program" or configure their system to perform even relatively simple tasks, such as turning the system on and off, or using a "universal" remote control (e.g., to watch or record a television program or videotape, laserdisc, or satellite broadcast). The advent of digital media and services (e.g., audio compact disks (CDs), direct satellite service (DSS) or digital video broadcast (DVB) from digital satellite broadcasts, and digital video disk (DVD) movies) creates a greater complexity and a need for a more general-purpose system or network solution.

A network distributing digital media and control information to devices has significant advantages. First, it is easier and less expensive to distribute information in a digital environment than it is to distribute and route analog audio and video streams to devices throughout a home or other environment, or process (e.g., record, mix, and otherwise modify) information along the way. Second, software control over individual "samples" of digital media provides significant new functionality that is not feasible in an analog domain. In a digital domain, all information (e.g., from audio, video, other digital media streams, custom commands, and asynchronous network protocols) can be distributed, processed, and controlled using the binary language spoken by virtually all of today's computer hardware and software, including dedicated hardware state machines, "smart" controllers, and general purpose central processing unit (CPU) based devices.

For example, in a digital domain users could decide to record a movie after watching the first 15 minutes, or record a song after hearing it in its entirety. Moreover, with a software interface, users can easily select a variety of programs to be recorded (e.g., over the course of a

week) without having to worry about switching tapes in a videocassette recorder (VCR). Additional telephone lines, or complex private branch exchange (PBX) functionality, could be integrated into the system, all under software control for greater flexibility. Playing and recording digital media streams can be as simple as reading and writing a disk drive, as will be explained below.

Current systems for interconnecting consumer electronics and other devices have a variety of problems. These problems include difficult and expensive cabling schemes, limited network topologies, complex and expensive devices, and inadequate media synchronization. There is also a lack of integration between the means for processing and controlling the information being distributed and the means for distributing audio, video, and other media.

A particularly important network parameter is the reliability of on-time delivery of time-sensitive data, sometimes called "quality of service." Quality of service in many prior-art networks is low, because many network protocols are collision-based (i.e., there is no handshaking or allocation of network resources between devices to preclude the need for repeat transmissions of time-sensitive data on congested networks). Audio, video, and other media are examples of time-sensitive data, and any network to distribute time-sensitive data must have a high quality of service in order to be practical.

### 3. Description of Related Art

#### 20 **Home Entertainment Systems and Home Automation Networks**

Current home entertainment systems and home automation networks illustrate many of the fundamental problems discussed above. For example, home theater systems effectively are limited to a single room in a house, primarily due to the difficulty and expense of interconnecting devices. Each source device, such as a satellite receiver, VCR, laserdisc, or DVD player, typically is connected, by relatively expensive audio and video cables, to a "central" preamplifier or other form of switching device, which is also connected to a main television/monitor as well as to one or more power amplifiers. These power amplifiers are also connected to various speakers throughout the room by dedicated (and also expensive) speaker cables.

Such a cabling scheme is expensive, obtrusive, and difficult to maintain, even in one room and especially in multiple rooms. Removing a defective VCR from a rack of equipment, for example, often requires reconnecting cables in a new configuration just to enable the rest



of the system to function while the VCR is being repaired. Moreover, flexibility is limited, even with relatively high-end preamplifiers. Only a limited number of source devices can be accommodated by any dedicated device interconnection scheme. It is often prohibitively expensive to replace a high-end preamplifier merely because it will not accommodate the latest source device added to the system. Users therefore still may suffer the inconvenience of connecting and disconnecting cables manually to switch sources.

Moreover, even when all devices can be accommodated, the functionality of such home theater systems still remains limited. Recording programs with a VCR, as noted above, remains a difficult objective. Users frequently must change tapes, as well as figure out how to “program” a remote control to make the preamplifier, source device, and VCR work together to record a program, particularly if someone is watching another program at the same time. Few systems even permit more than one program to be recorded simultaneously.

The difficulty of controlling consumer electronics devices and systems stems from the fact that most devices are not designed to operate with one another. User interfaces on universal remote controls are very limited. Liquid crystal display (LCD) screens, for example, are rarely used for more than listing a menu of commands. They rarely indicate any dynamic status information (e.g., which device is selected, which program is being recorded, and so forth). Moreover, universal remote control devices are extremely difficult to operate, and require extensive “programming” to perform even simple functions, such as turning the system on and off, which may require a “program” to send multiple commands (i.e., one command to each device). Even switching from one source to another may require a multiple-command program (e.g., a program to switch video input sources on the television as well as switch the preamplifier source).

The result is an expensive system that is complex to set up, use, and maintain in a single room. Multi-room systems are more rare, due to the expense of dedicated cabling solutions, and the additional inconvenience of distributing control over devices across rooms. If VCRs, CD players, and other source devices are physically located in different rooms of a house, the complexity of the “central” preamplifier, designed for dedicated cabling to all source and destination devices, becomes unwieldy. As a result, users improvise by providing redundant functionality. Multiple preamplifiers, multiple VCRs, and multiple remote controls (often “programmed” uniquely for each room) are deployed. This is not necessarily because multiple devices are needed at any given time, but primarily because the alternative is too

complex and expensive to be practical. Therefore, it is not surprising that such systems still are limited to hobbyists and high-end gadget-oriented users.

Current attempts to address the multi-room problem (i.e., "home automation and multimedia networks") exhibit a number of fundamental design flaws. For example, virtually all prior art home automation standards (e.g., X10, CEBus, Echelon LONWorks, and so forth) emphasize distribution of control information, but not audio, video, and other media. In other words, such systems distribute analog media on one network or wiring infrastructure and distribute digital control information on another separate network. Custom installation of dedicated audio, video, and speaker cables throughout a home is required, often is prohibitively expensive, and results in relatively poor quality due to the degradation of analog signals propagating through multiple devices and extremely long cable runs. Home power lines (or infrared or radio frequency broadcast signals) are typically used to form a separate digital control network onto which remote control/sensor devices can be connected.

The quality of such home automation networks is far below audiophile and videophile standards. Analog audio and video signals simply cannot propagate for long distances, and through multiple analog-to-digital (A/D) and digital-to-analog (D/A) converters and other analog processing circuitry, without suffering significant degradation in quality. Transmitting multiple audio/video channels over coaxial cables, power lines, or via radio frequency (RF) transmissions, degrades the signals significantly and results in relatively low-quality audio and video output.

Although some systems utilize existing unshielded twisted pair (UTP) telephone wiring to carry analog audio and video signals, the analog modulation of source audio and video signals over UTP cables also produces relatively low-quality audio and video output. Moreover, such systems still require custom installation of such UTP cabling, because the parallel nature of existing telephone wiring topologies (discussed in greater detail below) is not compatible with the dedicated cabling schemes discussed above. As a result, the expense and complexity of such systems increases exponentially compared with single-room home theater systems. Although control over devices is improved (e.g., by a personal computer (PC) or other controller device, operating under software or firmware control), it is still limited, in essence, to the commands accepted by individual devices (e.g., by their remote controls).

Home automation networks, as a result of their separation of control and media distribution functionality, do not allow for sufficiently flexible control over streams of audio,

video and other media, except at the most basic level of switching from one device to another. Existing home automation networks do not take advantage of the advent of digital media, continues to be distributed in analog form over a separate network, distinct from the control network that is the primary focus of such home automation networks. The result, as noted  
5 above, is a highly complex and expensive system that is difficult to set up, use, and maintain.

Furthermore, there is a convergence occurring between media and information data services (e.g., the Internet). Computer users are now recording television shows and using telephone transmission over the Internet. Cable television operators are now providing telephone and data services. Telephone companies are now offering data, video, and digital  
10 telephones. Therefore, the need for one integrated network to handle all of the converging media and information data services is rapidly increasing beyond the capabilities of prior-art networks.

#### **Local Area Networks and Ethernet**

One potential solution to the problems noted above is to implement a digital  
15 computer-based network of the type typically found in business environments for inter-connecting personal computers, workstations, servers, and printers. Although the recent popularity of the home computer market has led to a great deal of discussion of home networks, which merge general-purpose computing functionality with the consumer electronics devices found in home theaters, the solutions offered thus far have not advanced  
20 beyond the proposal stage.

Applying local area network technology (e.g., the Ethernet network protocol and the transmission control protocol/Internet protocol {TCP/IP}) to consumer electronics devices raises a number of problems. Although it appears advantageous to connect consumer electronics devices as generic nodes on a network, existing network protocols are not  
25 optimized for real-time streams of digital audio and video.

Prior art "solutions" typically fall into one of two categories. The first is analogous to the home automation networks discussed above, in which consumer electronics devices are connected via dedicated audio and video cables, and still transmit analog information along one network, while an Ethernet network, for example, enables home computers and other  
30 control devices to control the operation of the consumer electronics devices. The interface between the two networks is unclear, although the solutions offered by the X10 network standard and other home automation network standards appear to apply equally.

The other alternative implementation is to distribute audio, video, and other real-time media streams in digital form. There are a number of obstacles to this scenario, however, not the least of which is the absence of an existing physical and logical infrastructure to carry the digital media streams. Ethernet, for example, is not optimized to carry real-time continuous digital media streams. It is an asynchronous, packet-based protocol that would add significant overhead to digital audio and video samples, which require consistent and timely delivery, as opposed to the ability to send "burst" packets of information at high speeds on demand.

### FireWire

One present industry focus is a high-speed bus protocol, which was designed to enable personal computers to exchange information with peripherals such as disk drives, digital video camcorders, and digital televisions. This bus protocol, known as "FireWire" (described in the International Electrical and Electronic Engineer (IEEE) specification standard 1394), was designed as an information-exchange protocol to move data quickly between personal computers and peripherals. However, Firewire was not designed as a network protocol optimized for distributing real-time continuous digital media streams and control information among consumer electronics devices throughout a home.

FireWire devices are connected by a custom 6-wire cable, with two wires for power and two twisted-pairs for data. Devices can be daisy-chained to allow tree and other non-cyclic network topologies (i.e., no "loops" are allowed). On startup, the bus configures itself in three phases: bus initialization, tree identification, and self-identification. Whenever a new node is connected or disconnected, a signal sends all nodes into an initialization state, whereupon each node resets all topological information in a distributed manner. In other words, each node determines its own connections and passes this information along to its neighboring nodes, and so on throughout the network. One node is selected as the "root" node, and a unique physical ID is assigned to each node (though physical IDs are reassigned every time the bus is reconfigured).

FireWire employs two different processes, called "sub-actions," for distributing data packets among devices attached to the bus. One sub-action is asynchronous, while the other sub-action is isochronous. Asynchronous sub-actions begin with an arbitration period to determine which requesting device is granted control of the bus (because only one device can transmit at a time). The winning node transmits certain transaction and other codes (for speed, format, etc.), along with its address and that of destination nodes, and variable-length data. If the packet is not a broadcast packet, there is a brief gap, followed by an acknowledgment from

the destination node. Asynchronous sub-actions are followed by "sub-action gaps," which are required due to propagation delays.

5 Isochronous sub-actions are similar, but have a simpler arbitration process, a short channel identifier instead of source and destination addresses, no acknowledgment, and shorter gaps between sub-actions. FireWire uses a relatively slow 8 kHz clock for isochronous transmissions. After all desired isochronous packets have been sent, the bus resumes asynchronous sub-actions.

10 Communication between devices on a FireWire bus is half-duplex (i.e., devices connected to two or more nodes cannot propagate incoming data while transmitting their own data). FireWire devices employ a "data-strobe encoding" scheme (disclosed in U. S. Pat. No. 5,341,371), which involves transmitting data on one twisted-pair wire and a strobe signal on the other pair. The strobe signal transitions whenever two consecutive data bits are the same. This technique was believed to result in increased skew tolerance compared to a standard clocked format (such as 4B/5B or 8B/10B and clocked non-return to zero (NRZ) coding).

15 From the above summary of the characteristics of FireWire, it is apparent that there are a number of obstacles to applying FireWire to real-time distribution of digital media streams among consumer electronics devices, particularly in light of its essentially asynchronous and half-duplex nature. The installation cost of a FireWire network of consumer electronics devices is likely to be quite high. Custom installation of FireWire cables is usually required throughout a house, despite the use of twisted-pair cables. Moreover, no mechanism for compatibility with existing UTP telephone wiring topologies is included in the FireWire specification.

In addition to the high cost of installing FireWire, the cost of FireWire devices themselves is likely to be quite high, due primarily to the asynchronous nature of FireWire. 25 Because data packets are transmitted asynchronously, devices must be able to buffer incoming packets and generate timing information internally (i.e., "timestamps" associated with the data packets), which must be communicated to other devices. Moreover, the audio data rate can be almost doubled due to this buffering/time-stamping process, resulting in a significant loss of bandwidth. This results in complex and expensive devices, due to the memory, buffers, 30 counters, and associated circuitry required to perform such functions. Even a simple FireWire device requires at least a CPU and memory to implement the FireWire protocol stack, and the 8 kHz isochronous frequency is too slow for applications requiring low latency CD-quality (44.1 kHz) audio.

The lack of a truly synchronous time base also results in significant clock jitter. The beginning of an isochronous packet may be delayed by an asynchronous packet (due to their variable length). To account for this possibility, a delay is encoded in the "cycle start" signal, resulting in clock jitter. Moreover, additional jitter results from the variable amount of time  
5 between the "cycle start" and the actual start of a specific device's sub-actions, requiring each device to maintain its own timing register to account for this source of jitter. There are solutions to these timestamp and clock jitter problems, but these solutions add significant cost and complexity to every device attached to a FireWire bus, since every device needs a CPU just to manage these bus protocol issues.

10 For applications requiring real-time delivery of continuous digital media streams, FireWire has additional problems. One significant limitation to FireWire is the total length of the connecting cable, which is limited in the current specification to 4.5 meters (unless extremely expensive fiber optic or other alternative cables are used). This short length is a significant limitation within a single room used for a typical home theater system, and is  
15 prohibitive in the context of a house-wide application, such as a home telephone PBX or multi-room digital audio and video distribution system. Extensions to the Firewire standard now allow the network to be extended beyond 4.5 meters, but expensive protocol and media conversion boxes have to be installed on each end of a wire/fiber run. The network bandwidth for the whole system is limited by the low data rates possible over these other media.

20 An asynchronous FireWire router might become necessary to accommodate additional devices beyond this limit. Such a router would be extremely complex, requiring knowledge of data types to reassemble packets, buffering of variable-length packets, and so forth. In addition to being expensive, such a router would impose variable-length delays, making it extremely difficult, for example, to synchronize, with low latency, two speakers receiving  
25 information from a source device via the router.

The lack of static device identification (IDs) also creates problems. Each device must update its topology table whenever the bus is re-initialized (e.g., whenever a new device is added or removed, or a device malfunctions). Therefore, no device can rely on these device IDs remaining constant.

30 Because of the required transmission gaps between data transmissions on Firewire, it is very hard to utilize the full bandwidth that is possible on a particular media type. The half-duplex nature of Firewire further reduces the available bandwidth over a transmission medium by a factor of two. Since clocking and data are transmitted separately, two pairs of wires are

required for a connection to be established. These factors reduce transmission efficiency by a factor of eight when data is transmitted over Firewire.

### **Synchronous Networks and Ring Networks**

Unlike FireWire, a truly synchronous network could overcome many of the limitations discussed above. Yet, synchronous network protocols (as discussed below) have not been optimized or adapted for popular use with consumer electronics devices to enable the practical distribution of digital media streams.

Time-division multiplexed access (TDMA) networks, for example, utilize time-division multiplexing, and synchronize all devices to a master clock. However, the bandwidth on a TDMA network is typically divided equally among the devices on the network. In other words, if ten devices are on the network, each device gets one tenth of the network bandwidth, and thus can transmit information only during that channel or "time slice" (e.g., during one unit of every ten units of time).

However, in the context of transmitting digital media streams certain data streams require more bandwidth than others. For example, video data requires more data than audio, though it is sampled less frequently. Yet, TDMA networks assign each device a single channel in which to transmit all of its data. These channels are based simply on the number of devices on the network, and bear no relationship to the bandwidth requirements of the type of data being transmitted. This problem is exacerbated when asynchronously distributed variable bit-rate data, such as MPEG2 compressed video, needs to be accommodated. TDMA network technology provides no solution to either of these problems. TDMA devices are left with a single fixed-width channel that is not well-suited for accommodating either multiple real-time continuous data types having differing bandwidth requirements, or an asynchronous data type having bandwidth requirements that vary over time.

Even in the context of a ring network, in which data propagates from one device to another around a loop or ring (and is overwritten when a device desires to insert its own data), a device on a TDMA network could transmit information (such as a digital audio sample) anytime during its assigned time slice. Moreover, that time slice might change whenever a new device is added to or removed from the network. Thus, a device cannot guarantee consistent delivery of particular data, despite the synchronous nature of TDMA.

Ring networks, however, could be good candidates for consistently delivering particular data (e.g., a digital audio stream) from one device to another (e.g., from a CD player

to a speaker), assuming that such information propagates around the ring at a consistent rate (e.g., synchronously, based upon a master clock). Yet, existing ring networks suffer from many of the same deficiencies discussed above. Token Ring networks, as described in such specifications as the IEEE 802.5 Token Ring specification, are asynchronous in nature. Each  
5 device transmits data only when it receives the "token," which does not occur at fixed intervals of time. Thus, such networks cannot guarantee consistent delivery of real-time continuous digital media streams.

Fiber distributed data interface (FDDI) networks are similar to synchronous ring networks, but FDDI networks transmit information synchronously only in a point-to-point  
10 manner. In other words, the transmitter on one device is synchronized to the receiver on the next device on the ring, but the transmitter and receiver within a device are not synchronized to each other. Therefore, information will not always propagate through a device at a consistent rate, due to the difference between the transmit oscillator and receive oscillator within a device, among other factors.

FDDI devices compensate for this difference with an "elasticity buffer" which avoids  
15 losing data, but this does not guarantee consistent delivery of data. For example, if a device receives data "late," it will transmit that data late. If it receives data "early," it will place that data in its elasticity buffer, and transmit such data in a first-in-first-out (FIFO) fashion. Thus, FDDI devices also cannot guarantee consistent delivery of data such as real-time continuous  
20 digital media streams. They are optimized for high throughput, but not for consistent, synchronous delivery of data.

Both TDMA and FDDI systems do not transmit bi-directional data over a single pair of wires. This means that multiple pairs of wire must be run between devices, or the network must always be wired in a physical loop configuration.

## 25 **Resolving the Problem**

The above description of home theater systems and home automation networks, and of various existing asynchronous and synchronous network protocols, illustrates many of the obstacles to interconnecting consumer electronics devices for distribution of audio, video, other media (including both real-time continuous digital media streams and asynchronous  
30 data), and control information throughout a home or other environment. These obstacles must be resolved before home networks can achieve widespread acceptance. What is needed is a low-cost physical network topology, preferably one that is compatible with the existing physical cabling infrastructure in a home to avoid the significant barrier of having to rewire an



entire home. Such a network topology should enable devices to be interconnected with relative ease, compared to the difficulty and expense of interconnecting audio and video and other consumer electronics devices within and across rooms of a home using prior art technologies.

5           Such a network also should be compatible with existing consumer electronics devices, again to avoid the significant problem of having to replace all existing devices merely to set up a simple network consisting of a few devices. Moreover, such adapters and other devices preferably should be relatively inexpensive (at least no more expensive than comparable existing devices) in order to encourage consumers to adopt this new technology.

10           The network should provide a network that can accommodate real-time continuous digital media streams (e.g., digitized audio, video, and telephone), as well as asynchronous data and traditional data networking protocols. To do so, the network must deliver digital media streams reliably with a high quality of service, in order to provide the same level of synchronization as is currently provided by existing analog delivery mechanisms.

15

#### SUMMARY OF THE INVENTION

One object of the present invention is to provide a low-cost physical network topology compatible with the existing physical cabling infrastructure in a home.

20           Another object of the invention is to provide a network compatible with existing consumer electronics devices.

Another object of the invention is to provide a network that can accommodate real-time continuous digital media streams (e.g., digitized audio, video, and telephone), as well as asynchronous data and traditional data networking protocols.

25           The present invention fills these needs by providing a protocol and architecture for a synchronous logical ring network which operates on the existing physical twisted-pair telephone topologies found in most homes today (forming a "logical" ring without requiring in-wall wiring modifications). The present invention can be implemented in numerous ways, such as a method, a system, an apparatus, and a program on electronic-readable media. Several aspects of the invention are described below.

30           In accordance with a first aspect of the invention, the invention provides a method and apparatus to communicate information by using symbols generated by a source device on a network for transmission in the network by the source device and reception in the network by

a destination device. The method includes encoding the information to produce an encoded symbol; expressing the encoded symbol as a scrambled multi-level electrical signal at the source device, wherein the multi-level electrical signal has at least three levels; de-scrambling the scrambled multi-level electrical signal at the destination device to determine the encoded symbol; grouping at least one of the encoded symbols into a symbol group; and decoding the symbol group into the information.

In accordance with a second aspect of the invention, the invention provides a method and apparatus to transmit a command stream generated by a source device for reception by a destination device in a network connecting a plurality of devices. The method includes appointing one of the devices as a clock master to provide a command stream token on the network, wherein each source device on the network is required to receive the command stream token before transmission of the command stream, and hold the command stream token until transmission of the command stream is complete; releasing the command stream token to another device on the network; encoding the command stream as a group of one or more symbols, wherein each symbol is represented by a multi-level electrical signal having at least three levels; receiving at the destination device one or more multi-level electrical signals representing the group of one or more symbols; and decoding the group of one or more symbols to reconstruct the command stream at the destination device.

In accordance with a third aspect of the invention, the invention provides a method and apparatus to transmit an audio stream generated by a source device on a network for reception in said network by a destination device. The method includes expressing the audio stream as a group of one or more symbols, wherein each symbol is represented by a scrambled multi-level electrical signal having at least three levels; de-scrambling and error correcting the scrambled multi-level electrical signal at the destination device to determine the group of one or more symbols; grouping one or more symbols into a symbol group to be decoded; and decoding the symbol group to reconstruct the audio stream at the destination device from at least two of the symbol groups.

In accordance with a fourth aspect of the invention, the invention provides a method and apparatus to transmit an asynchronous packet stream generated by a source device on a network for reception in the network by a destination device. The method includes expressing the asynchronous packet stream as a group of one or more symbols, wherein each symbol is represented by a scrambled multi-level electrical signal having at least three levels; de-scrambling the scrambled multi-level electrical signal at the destination device to determine

the group of one or more symbols; grouping said one or more symbols into a symbol group to be decoded; and decoding the symbol group to reconstruct the asynchronous packet stream at the destination device from at least two of the symbol groups.

In accordance with a fifth aspect of the invention, the invention provides a method and apparatus to transmit a telephone stream generated by a source device on a network for reception in the network by a destination device. The method includes expressing the telephone stream as a group of one or more symbols, wherein each symbol is represented by a scrambled multi-level electrical signal having at least three levels; de-scrambling the scrambled multi-level electrical signal at the destination device to determine the group of one or more symbols; grouping one or more symbols into a symbol group to be decoded; and decoding the symbol group to reconstruct the telephone stream at the destination device from at least one of the symbol groups.

In accordance with a sixth aspect of the invention, the invention provides a method and apparatus to determine a clock offset on a logical ring network having a plurality of devices. The method includes synchronizing a plurality of frame-counting clocks on the network by broadcasting a time mark command; specifying a frame count for each device of the plurality of devices that needs its frame count synchronized with other devices of said plurality of devices on the logical ring network; following the time mark command with the transmission of a marked frame that goes around the logical ring network; calculating a time difference value from the time mark command and the marked frame for at least one device; and transferring the time difference value into a frame-counting clock in at least one device, wherein the time difference value is used to calculate a clock offset for at least one device.

In accordance with a seventh aspect of the invention, the invention provides a method and apparatus to interface stream information between one or more network control protocols and a network physical layer. The method includes processing command and data in the stream information with a command stream processor; generating one or more network time and event signals with a network time and event generator; reading and writing command and data in the stream information communicated to the network physical layer on a physical layer interface; reading and writing serial data provided on a serial memory interface; and selectively resetting the command stream processor, the network time and event generator, the physical layer interface, and the serial memory interface.

In accordance with an eighth aspect of the invention, the invention provides a method and apparatus to elect a device as a clock master from a plurality of devices on a logical ring

network. The method includes selecting a first device of a plurality of devices as the clock master using an arbitration value; sending a first message from the first device, wherein if the first device receives an acknowledgment on all of the ports, the first device is elected the clock master, but if the first device receives a message from a second device containing a higher  
5 arbitration value than the arbitration value of the first device, the first device sends a second message containing the higher arbitration value of the second device out on all remaining ports; and sending a third message to the second device with the higher arbitration value as an acknowledgment message appointing the second device as the clock master, if an  
10 acknowledgement is received on all of the other ports of the first device, or sending the third message to the second device with a higher arbitration value if a message with an even higher arbitration value than the arbitration value of the second device arrives on any of the ports of said first device.

In accordance with a ninth aspect of the invention, the invention provides a method and apparatus to allocate a set of lanes in a frame containing a plurality of lanes, in a network  
15 connecting a plurality of devices. The method includes transmitting a value from an originating device requesting the set of lanes in the frame, wherein the set of lanes are represented by a plurality of bits in the value; receiving the value at each device of the plurality of devices; removing a bit from the value for each corresponding lane of the plurality of lanes that each device is using; receiving the value at the originating device; and setting a  
20 set of bits in a mask representing the set of lanes at the originating device, wherein the set of bits reserves the set of lanes for the use of the originating device.

In accordance with a tenth aspect of the invention, the invention provides a method and apparatus to transmit bi-directional synchronous data streams from a first device to a second device on a time-division multiplexed access (TDMA)-oriented network connecting a plurality  
25 of devices. The method includes allocating a first set of lanes in a frame containing a plurality of lanes to the first device when the first set of lanes can be allocated, wherein the frame is received by the second device; allocating a second set of lanes in the frame to the second device when the second set of lanes can be allocated, wherein the frame is received by the first device; transmitting a first group of synchronous data on the first set of lanes from the first  
30 device to the second device; and transmitting a second group of synchronous data on the second set of lanes from the second device to the first device.

In accordance with an eleventh aspect of the invention, the invention provides a method and apparatus for broadcasting device identification during startup of each device in a network connecting a plurality of devices. The method includes sending status information

from each device to the remainder of the network connecting the plurality of devices; and sending device configuration information from each device to the remainder of the network connecting the plurality of devices.

In accordance with a twelfth aspect of the invention, the invention provides a method and apparatus for structuring the data architecture of a device read only memory (ROM) in a network connecting a plurality of devices. The method includes assigning a first plurality of bytes in a device ROM of the device for one or more of the following purposes selected from the group consisting of: Protocol Version Number, Company ID, and Model ID; assigning a second plurality of bytes in the device ROM of the device for one or more of the following purposes selected from the group consisting of: Protocol Hint Bit Mask, Capability Hint Bit Mask, and the number of streams implemented by the device; and assigning a third plurality of bytes in the device ROM of the device for one or more of the following purposes selected from the group consisting of: start address of the device's Allocation Marker in a random access memory, start address of the device's Device Information in ROM, device status, logical loop number, and loop operating mode.

These and other objects and advantages of the invention will become apparent to those skilled in the art from the following detailed description of the invention and the accompanying drawings.

### BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objectives, aspects, and advantages will be better understood from the following detailed description of embodiments of the present invention with reference to the following drawings:

FIG. 1 illustrates a typical system-level configuration of consumer electronics devices connected to one preferred embodiment of the invention in a typical home.

FIG. 2 shows an example of a MAC interface illustrating how the speakers in FIG. 1 are configured.

FIG. 3 shows another example of a MAC interface illustrating one preferred embodiment of how a device with a central processor (CPU) is configured with the MAC interface.

FIG. 4 shows a detailed diagram of one preferred embodiment of a MAC chip.

FIG. 5 is a diagram illustrating the configuration of the MAC interface shown in FIG. 1.

FIG. 6 illustrates decoding physical symbols into a frame in one embodiment of the invention.

FIG. 7 illustrates one embodiment of the invention with lane assignments for three streams.

5        FIG. 8 illustrates devices that write, read, modify, or pass-through streams.

FIG. 9 illustrates one preferred embodiment of a frame header and command lane assignments.

FIG. 10 illustrates one example of creating a logical loop from a daisy-chain.

FIG. 11 summarizes the top-level states and state transitions of a device.

10       FIG. 12 illustrates an optional mechanism to enhance the accuracy beyond the standard clock synchronization.

FIG. 13 illustrates the most preferred embodiment for encoding on the wire and at the PHY and MAC layers.

15       FIG. 14(a) and 14(b) illustrate network 5-level data code pairs according to one embodiment of the invention.

FIG. 15 illustrates 5-level control code pairs according to one embodiment of the invention.

FIG. 16 shows auto-negotiation can be broken up into several major phases: link fail, clock arbitration, speed arbitration, loop configuration, and normal operation.

20       FIG. 17 shows the state machine for half-duplex communication for one preferred embodiment.

FIG. 18 illustrates one example of coding of a 20-bit sample on an audio stream, and shows how a stream sample consisting of a single 20-bit audio sample is formatted and distributed onto lanes in a frame.

25       FIG. 19 shows how a stream sample consisting of a stereo pair of 20 bit audio samples would be formatted and distributed onto the frame.

FIG. 20 depicts four stations on a ring and illustrates the four basic steps of asynchronous packet stream operation; token circulation, transmitting, receiving, and stripping.

30       **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

Alternative embodiments of the invention can be implemented with devices other than consumer electronics devices (e.g., business devices, industrial devices, professional devices, and so forth) and in environments other than a home (e.g., offices, hotels, apartment buildings, cars, boats, recreational vehicles, and so forth). The particular environment or application may  
5 result in a significantly different configuration than the configurations illustrated below. The advantages of the embodiments of the invention described below with reference to a network of consumer electronics devices in a home also apply to these other environments and applications.

A logical ring network can be as simple as connecting two network devices together  
10 with a standard RJ-11 telephone cable (or standard or slightly modified RJ-12, RJ-45, or other connectors in other embodiments). More devices can be added to the network by daisy-chaining them directly to one of these two devices, or connecting them to a junction box (for distribution throughout a home). All devices and/or components illustrated are shown attached to logical ring network in a daisy-chain fashion (either within a room or behind the  
15 walls via a junction box). The actual physical topology of a logical ring network, and the manner in which devices are interconnected, is discussed extensively in co-pending U. S. Patent Application Serial No. 09/079,914, entitled "Synchronous Network for Digital Media Streams," filed on May 15, 1998, which is incorporated by reference.

Illustrated in FIG. 1 is a typical system-level configuration of consumer electronics  
20 devices connected to one preferred embodiment of the invention in a typical home 100 having a first room 102, a second room 120, and a third room 146.

The first room 102 has devices connected in a daisy chained fashion including an incoming cable line 104, a set top box 106, a DVD player 108, a CD player 110, a stereo 112, a telephone 114, five speakers 116, and a television monitor 118. Each device is connected  
25 through a media access controller (MAC) and physical (PHY) interface (not shown for set top box 106, DVD player 108, CD player 110, and stereo 112, to reduce drawing complexity). The daisy chain continues into the second room 120.

The second room 120 includes telephone adapter 122, telephone 124, stereo 126, speaker 128, television monitor 130, speaker 132, speaker 144, MAC interface 134, and PHY  
30 interface 135. MAC interface 134 and PHY interface 135 can be implemented in one or more integrated circuit chips. MAC interface 134 is connected to converters and relays that are in turn connected to camera 136, microphone 138, light 140, and doorbell 142. MAC interface 134 is also connected to a physical layer (PHY) interface 135 that is connected to the

MAC/PHY interfaces of telephone 124, stereo 126, and stereo 144. Telephone 124 can be a digital telephone, capable of accessing the legacy analog telephone network via a plain old telephone service (POTS) version of telephone adapter 122. Such devices can have built-in PBX functionality, or can provide such functionality via a general-purpose DSP device (not shown).

The third room 146 includes a DSS receiver 148 with a MAC/PHY interface connected to the network in second room 120. The DSS receiver is connected to a satellite dish 150 to receive satellite signals. DSS receiver 148 receives broadcast audio/video information (e.g. MPEG2). Because MPEG2 information already is in digital form and compressed, it can be transmitted directly onto the network. As will be explained below for one embodiment, the MPEG2 decoding will occur only after such information leaves the network (e.g., at a television where the information will be decoded and decompressed for viewing).

Information transmitted by these source devices propagates around the network and can be received by any appropriate destination device, as will be described in greater detail below. For example, an MPEG2 movie, received by DSS Tuner 148 or played at CD/DVD player 110 or 108, could be displayed on television monitor 118 and/or television monitor 130 after being decoded by MPEG2 decoders (not shown). The audio extracted by an MPEG2 decoder could be transmitted back onto the network and played by speakers 116, 128, 132, or 144. The audio extracted by a MPEG2 decoder could be transmitted back onto the network, processed by digital signal processing (DSP) (e.g., an AC-3 surround sound decoder), and then transmitted back onto the network and played by surround sound speakers 116. In alternative preferred embodiments, remote control/interface/monitor devices (not shown), such as simple remote controls or complex controllers with large LCD screens capable of viewing computer graphics images, MPEG2 video, and so forth, also control the systems in rooms 102, 120, and 148.

As will be discussed below, virtually any information obtained or generated by one network device can be distributed "simultaneously" (i.e., within the same digital sample, appropriate to the media type) to any other network device, because such devices are connected to the network. For example, DVD 108 in first room 102 might be playing various digital audio/video programs previously selected by a family member. Upon request, such information can be transmitted along the network for immediate viewing, or saved for later viewing (e.g., to avoid being overwritten as new programs are played).



This embodiment of a home network provides a great deal of flexibility. Not only can any network device distribute real-time continuous digital media streams (and possibly asynchronous data) to any other network device, but such devices (including legacy devices connected via an adapter, as well as new "digital-ready" devices) can exhibit functionality not feasible with existing technologies, such as recording television programs already in progress in their entirety. It is significant to note that network devices need not be complex or expensive to perform basic functions such as communicating with the network and transmitting and receiving digital media streams. Other embodiments of the invention can be implemented by adding devices of greater complexity, particularly CPU-based devices controlled by software or firmware (as will be discussed below).

FIG. 2 shows an example of a MAC interface 200 illustrating how the speakers in FIG. 1 are configured. MAC interface 200 has two input/output plugs 202 and 204 connected to PHY 206. PHY 206 is connected to MAC chip 208. The serial audio input/output of MAC chip 208 is then connected to a digital-to-analog (D/A) converter 210 that is connected to an amplifier 214 and speaker 212.

FIG. 3 shows another example of a MAC interface 300 illustrating one preferred embodiment of how a device with a central processor (CPU) 310 is configured within MAC interface 300. MAC interface 300 has two input/output plugs 302 and 304 connected to PHY 306. PHY 306 is connected to MAC chip 308. The host bus connection on MAC chip 308 is then connected to CPU 310, where CPU 310 can control any components (not shown) that are connected to bus 312. PHY 306 implements all of the line adaption and line coding, equalization, echo cancellation, time base derivation, and so forth. In one embodiment of the invention, there are 3 bit wide parallel digital signals being sent in both directions between PHY 306 and MAC chip 308.

FIG. 4 shows a detailed diagram of one preferred embodiment of a MAC chip 400. MAC chip 400 has a number of components to control the incoming and outgoing digital data. MAC chip 400 includes command stream processor 402, stream engines 404, network time & event generator 406, PHY interface 408, serial memory interface 410, reset circuit 412, media asynchronous packet stream (MAPS) 414, two synchronous direct memory access (SDMA) circuits 416, two video stream encoder/decoders 418, four audio stream encoder/decoders 420, general purpose digital I/O (GPIO) block 422, and host CPU interface 424.

In one preferred embodiment of the invention, command stream processor 402 utilizes four lanes of data on the network for a native command packet protocol that implements

network control, as well as the allocation of network resources and devices on the network. The number of lanes assigned could be less than four or greater than four in other preferred embodiments. Additionally, command stream processor 402 controls primitive devices if they do not have a host processor, wherein a processor on another node can control the device by using commands to read and write registers using the device.

In one preferred embodiment of the invention, a frame of data is sent on the network every 20.83 microseconds. Each frame of data is divided up into a number of lanes. Several modes of data transmission are possible and the selected mode determines the number of lanes in a frame. At the beginning of a frame there are four lanes dedicated to a frame marker for synchronization, and four lanes dedicated for a command stream to be used by command stream processor 402. The remaining lanes in the frame are available for dynamic allocation to pass data between devices on the network.

Stream engine 404 maintains the timing of the frames and informs any interfaces that are getting data from the frames when their lanes appear in each frame. In preferred embodiments, there is a stream engine 404 for each interface on the network. Network time & event generator 406 has all of the clocks, the offset for one or more clocks, and participates in controlling events that need to use clocks having offsets for increased precision of synchronization between devices. PHY bus 426 transfers data streams to and from the network via PHY interface 408.

Serial memory interface 410 is used to add external read only memory (ROM). This is preferably done either with an external interface chip, or as programmable ROM on the chip. This feature allows configuration of hardware and device ROM so that other devices can see what kind of device this is, and what this device's capabilities are. In one preferred embodiment of the invention, the identification (ID) number for each chip is stored on an internal ROM, or a ROM accessible through serial memory interface 410.

Reset circuit 412 provides reset logic to selectively reset MAC chip 400 when the network resets. In preferred embodiments of the invention, it can reset the external hardware and external hardware can reset it.

MAPS 414 provides facilities for handling asynchronous packets on the data stream lanes. The network is a synchronous network, therefore the data can be optimized to connect to the Ethernet and carry Ethernet packet traffic. The resulting data packets travel through host bus 428.

SDMA 416 allows taking some set of lanes, or an entire frame and make a direct memory access (DMA) into the memory or the host bus. Two SDMAs 416 are provided, one of them to output data to memory and one to input data from memory.

Video stream encoder/decoders 418 handle any incoming or outgoing compressed digital audio or video signals through a corresponding video bus 430. Audio stream encoder/decoders 420 handle uncompressed data (which is not packetized), and compressed structured data, and sends or receives digital audio signals through serial audio connection 432.

GPIO block 422 has eight interface lines that are one bit digital input/output interfaces. If used as input pins, they can be programmed to trigger the device to broadcast a message via the command stream processor 402 when the state changes.

Host CPU interface 424 in preferred embodiments appears as parallel ports to an external processor. It requests DMA transfers (up to six channels of DMA in one preferred embodiment), generates interrupts for the host, and supports an interrupt structure. If there is no external processor, then an external memory or device module can be added using these address and data lines.

FIG. 5 is a diagram illustrating one possible embodiment of the MAC interface 134 and PHY 135 interface shown in FIG. 1. Video camera 136 is programmed to pass its video signals to an encoder 502, and in turn the digital video signals are passed to video bus 504 on MAC interface 134. In one preferred embodiment of the invention, digital audio can optionally be sent from MAC interface 134 along serial audio connection 506 to digital-to-analog (D/A) converter 508, and played over speaker 144, or sent directly to the speaker 144 MAC/PHY interface (shown in FIG. 1). Microphone 138 passes its audio signals to analog-to-digital (A/D) converter 510, and to MAC interface 134 through serial audio connection 512. Light 140 is toggled on and off through MAC interface 134 from GPIO pins 514. When doorbell 142 is activated, a toggled switch passes information through GPIO pins 514.

As an example of the system shown in FIG. 5, a person 190 (shown in FIG. 1) approaches house 100 and pushes doorbell 142. Activated doorbell 142 triggers the command stream processor (shown in FIG. 4) to send a message on the network through the PHY interface (shown in FIG. 4) indicating that doorbell 142 has been activated. The manager of the network, such as a computer program (not shown) passes instructions back to MAC interface 134 instructing it to turn light 140 on, turn video camera 136 on, and turn microphone 138 on. The video and audio data are selectively sent to any video/audio monitors

within the house. A person (not shown) in house 100 sends audio signals through the network to play on speaker 144 to person 190.

### **Synchronous Logical Ring Network**

Network devices on a logical ring network of a preferred embodiment of the invention  
5 are synchronized to one another, and each imposes a constant amount of delay (differing from device to device) to process signals. Given that signals traveling along the network path encounter each network device exactly once per "revolution" around the entire logical ring network, the signal propagation time around the logical ring network along this network path (and between any two particular devices along the way) remains constant. As a result,  
10 network devices are afforded a fixed amount of total network bandwidth for sending and receiving information along this network path. Various means of increasing network bandwidth will be discussed in greater detail below.

The wires may be of a different type from standard residential unshielded telephone wiring. Category 3 or Category 5 unshielded twisted pair wiring, or even coaxial or fiber optic  
15 cable, may be installed in order to support higher-speed signal propagation, and thus provide increased cable lengths and/or network bandwidth. As will become apparent, the logical ring network of preferred embodiments of the invention can support virtually any transmission media, including wireless configurations.

A logical ring network (or an extension of an existing logical ring network) is created  
20 by daisy-chaining devices to one another. It is not necessary to connect any devices to a junction box or perform any wiring modifications. For example, a daisy-chain of cables to a room extends the logical ring network to the room. Alternatively, the wiring at a single junction box is modified in each room to connect the room to the logical ring network. Thus, a logical ring network can be extended gradually throughout a house to add new devices as the  
25 need arises.

In an alternative embodiment, any wiring segment between two devices could be replaced with a bi-directional wireless link (e.g., RF, infrared, ultrasonic, or equivalents). For example, by employing wireless transceivers at the two endpoints of a chain of devices having bi-directional transceivers, a logical ring network could be constructed with the wireless link  
30 serving to complete the default network path.

In an alternative embodiment, an entire network is configured with the capability of switching between an analog telephone network and a logical ring network. In such a

configuration, however, in order to complete the default network path of the logical ring network, the devices would require bi-directional transceivers.

It should be noted that, using standard unshielded twisted pair wiring, whether in a star or other physical topology, the total distance between network devices along the network path  
5 may have a practical limitation due to signal attenuation. This limitation will depend on the quality of the wire. Standard "silver satin" wire, for example, may be limited to approximately 100 feet, whereas Category 5 wire may perform reliably with runs in excess of 300 feet. Should this limitation arise, devices that act as repeaters, to amplify the signal, can be added to the logical ring network.

Moreover, in the most preferred embodiment, a "smart jack" is provided which not only closes the physical loop regardless of whether a network device is connected, but also maintains a fixed amount of network delay. Such a mechanism enables one to connect and disconnect network devices (daisy-chained to the smart jack) without disrupting the transmission of information (e.g., an audio or video stream) among other devices on the  
10 logical ring network. This "hot pluggable smart jack" is itself a network device into which a chain of other network devices can be connected, but which adjusts its delay (and causes the logical ring network to be reconfigured) to account for the insertion or removal of any of such other network devices without disrupting the transmission of information among the remaining network devices (including information passing through the hot pluggable smart jack itself).  
15 This embodiment and the other embodiments discussed below, are described in greater detail in co-pending U. S. Patent Application Serial No. 09/079,914, entitled "Synchronous Network for Digital Media Streams," filed on May 15, 1998, hereby incorporated by reference.  
20

In yet another embodiment, it may be desirable to enable a homeowner or device modules to manually or automatically switch between the conventional POTS analog  
25 telephone network and the logical ring network of a preferred embodiment of the invention, merely by manually or automatically toggling a switch at the junction box and in a network interface adapter for an analog telephone in emergencies or when the network is unable to provide networked telephone service. It should be evident to one of ordinary skill in the art that there are many mechanisms for switching the wiring infrastructure from a POTS network  
30 to the logical ring network configuration of a preferred embodiment of the invention, including removable modules/printed circuit boards, reversible boards, relays, and so forth. As will be discussed below, it is even possible for a POTS network to coexist (i.e., to share a pair of wires) with a logical ring network.

A single pair of wires can support both a logical ring network (with bi-directional drivers) and a POTS network. This is accomplished by using high-pass and low-pass filters, respectively, at each wiring junction to isolate each of these types of signals from one another. Such filters are "high-frequency balanced" to prevent the high-frequency network signal from becoming unbalanced, and thereby creating radio interference.

Multiple types of signals can share the same pair of wires, which can be unshielded twisted pair, coaxial, or other transmissive media. If another pair of wires were available, it could be dedicated to a logical ring network and/or power, leaving the POTS network untouched on its own existing pair of wires, thereby preventing any interference between the phone system and the power lines or a logical ring network. Alternatively, one pair of wires could be dedicated to the bi-directional logical ring network, with the other pair supporting the POTS network or power.

#### **Logical Ring Network**

The most preferred embodiment is based on a logical ring topology, which is synchronous and stream based. A stream is a constant bandwidth flow of information carried on one or more lanes. The network supplies multiple streams simultaneously.

In the most preferred embodiment, the basic transport protocol is a TDM scheme in which data frames (frames) propagate around the network at the rate of 48,000 frames per second. Each frame contains a set of equally sized time slots ("lanes") the number of which depends on the signaling rate (the network operating mode) in use at the physical level. The data rate of a stream depends on the number of lanes assigned to carry the stream.

At network start up, devices arbitrate for the role of "Clock Master." In less than a second, one device is elected as Clock Master. From that point on, all other devices will slave their time base to that of the Clock Master. Synchronous operation greatly simplifies the operation of devices that handle isochronous or time-critical data.

#### **Media Access Control**

As is the case with most networks, the total available bandwidth on the preferred embodiment is more than any single device requires. The media access control (MAC) protocol governs how devices access and share the available bandwidth. Unlike many other networks, media access is not collision based, and data is not organized into packets. Bandwidth for a stream is reserved and guaranteed in terms of sets of windows on a frame.

At the signaling level, each frame is a progression of symbols. In the most preferred embodiment, each symbol has one of 5 possible values, but alternative embodiments could use a greater number (e.g., 7) or lesser number (e.g., 3) of values. In the most preferred embodiment, symbols are grouped together in sets of five, called "quintets," but alternative  
5      embodiments could use a greater number (e.g., 6 or more) or lesser number (e.g., 2 or more) of symbols in a group. A quintet carries two lanes worth of information. Each quintet in this embodiment (termed 9B5Q) encodes eight data bits, four data bits and a control code, or two control codes, i.e., the quintet carries more than one byte of information.

FIG. 6 illustrates decoding physical symbols into a frame 606 in one embodiment of  
10      the invention. In a preferred embodiment, frame header 604 occurs at the beginning of frame 606 and occupies two quintets 602. It is a unique code that is forbidden to occur elsewhere in the frame. Frame 606 is defined as a frame header and all of the subsequent quintets transmitted until the next frame header. Frame headers arrive at a given device with a very regular and accurate period, 20.83  $\mu$ sec (1/48 kHz). The frames propagate all the way around  
15      the logical loop. Therefore, frame headers arrive at every device with the same frequency and accuracy. However, a given frame header will arrive at each subsequent device slightly after its arrival at the previous device, due to the delay introduced by the regeneration of the signal at every device, and the delay caused by the length of wire connecting the devices.

Each device uses the frame header to initialize an index count, which is then used to  
20      partition the data into the lanes that follow. As mentioned above, each lane transports a progression of data nibbles, interspersed with any of several control or escape codes. Once the network has initialized and entered normal operation, all frames on a given loop contain the same number of lanes. The number of lanes per frame is determined by the speed of the network, i.e., the symbol rate.

25      A stream is a constant bandwidth flow of data between two points on the loop. It is carried by a fixed set of lanes, termed the stream's lane assignment in the loop. As each frame passes by a device, this set of lanes is accessed in order to access the stream. The set may be comprised of lanes at arbitrary positions within the frame. This mechanism is similar to many time-division multiplexing (TDM) schemes. The lanes assigned to a particular stream need  
30      not be contiguous. They may be scattered throughout the frame, as shown in FIG. 7. This allows complete flexibility when it comes to creating new streams. It also avoids the fragmentation problems that would otherwise result after the network had been in operation for a while.

FIG. 7 illustrates one embodiment of the invention with lane assignments for three streams, stream 1 702, stream 2 704, and stream 3 706. The abstract capability of a device to read, write, or modify a single stream is called a "device stream." The implementation of that capability at the MAC level is called a "stream engine." Devices may support up to 255 device streams, i.e., it is possible to specify a device that can read, write or modify 255 separate streams simultaneously.

In preferred embodiments of the invention, a command is available for negotiating lane allocations and other peer-to-peer privileges among smart devices hosting processes that need to control network resources. Devices can make claims for specific resources (e.g., the lanes in a given bank of lanes in a frame). Bits in a selected register or mask of each device can represent the current status of the same resources. As a claim circulates, the claim is revised to exclude the resources already marked as committed at other devices. When the claim returns to its sender, the sender determines what resources were successfully acquired. Specific resources, such as a number of lanes in a frame, can be released for use by other devices by indicating their availability by appropriate bits in the selected register or mask of each device.

The most preferred embodiment of the invention allocates a set of lanes in a frame containing a number of lanes, among a network of devices by first transmitting a value from a device requesting a set of lanes in each frame, wherein the set of lanes are represented by bits in the value. Each of the other devices in the network receives the value and removes bits corresponding to lanes that the other device is using from this value using real-time processing as the command goes through the device. The requesting device receives the value, and sets the bits in a mask representing the lanes it requests and reserves to use.

Each device may separately handle each stream on a loop, while passing the stream through, reading it, modifying it, or writing it. Each device not modifying or writing the stream regenerates the stream from its input to its output. FIG. 8 illustrates devices that write, read, modify, or pass-through streams. The stream-based architecture allows devices to operate in one of 4 modes: writer, reader, modifier, or pass-through. For example, in FIG. 8 device A 802 is a writer of Stream 1. The stream is available at any point around the loop, but in this case Device C 806 is the only reader. Device C 806 does not overwrite the stream as it reads it, and the stream returns unmodified to Device A 802. Devices B 804 and D 808 allow Stream 1 to pass through, unmodified and unread. Stream 2, on the other hand, is read and modified by both Device B 804 and D 808. In most cases, modifier devices introduce a one-



frame delay in the stream being modified. A modifier device reads the stream contents and then overwrites the stream with new data.

### Throughput

The total number of lanes assigned to a stream determines its bit rate. This assignment  
5 occurs when a stream is created and remains fixed for the life of the stream. In the most preferred embodiment, a slow stream has only a single lane assigned to it and delivers information at a bit rate of 192 kbps. The fastest stream would use every lane in the frame, minus 8 lanes that are dedicated for network support and would deliver data at a bit rate of several tens of Mbps. The exact maximum rate depends on the network operating mode, i.e.,  
10 on the number of available lanes in the frame. In practice, any given stream needs only a portion of this total network capacity.

In a preferred embodiment of the invention, one operating mode could appear to have a total throughput of about 49 Mbps. However, this is deceptive because on a network according to a preferred embodiment, several devices may modify the same stream.  
15 Therefore, the basic bit rate of a stream must be multiplied by the number of writers or modifiers of the stream in order to arrive at the stream's total throughput.

For example, an audio processing chain consisting of a CD player, an audio DSP, and several digital audio amplifiers can be implemented on a preferred embodiment using a single stream, provided that the devices are physically connected to the network in the correct order.  
20 On an IEEE 1394a bus each modifier device in a signal processing chain requires a separate stream for its output, and each of these streams counts separately against the total throughput of the network.

Comparing the throughput of networks with asynchronous transport protocols, such as IEEE 1394a, is further complicated because asynchronous transport imposes a substantial  
25 overhead on streaming media in the form of embedded time-stamps required for re-synchronization. Networks according to a preferred embodiment of the invention have no corresponding overhead for streaming media because they are inherently synchronous.

To provide a rough handle on network throughput the network operating modes have been labeled according to their approximate throughput in the full-duplex case, i.e. where each  
30 stream has two modifiers. That is why networks running at a basic rate of about 50 Mbps are said to be operating in a MW 100 mode. In this operating mode, two devices could simultaneously transmit to each other using all available lanes) at roughly 49 Mbps, for a total

network throughput of 98 Mbps. Table 1 lists the operating modes in one preferred embodiment, but other operating modes could be used in alternative embodiments.

Name	Symbol Rate /Sec	Bit Rate	Lanes/ Frame	Overhead Lanes	Efficiency
MW25	7.68 M	12.288 Mbps	64	8	88%
MW50	15.36 M	24.576 Mbps	128	8	94%
MW100	30.72 M	49.152 Mbps	256	8	97%
MW200	61.44 M	98.304 Mbps	512	8	98%

Table 1 - Network Operating Modes

## 5 Frame Format

In the most preferred embodiment, the period of frame transmission is always constant at 20.83  $\mu$ sec (i.e. the frame arrival frequency is 48 kHz). The different loop bit rates produced by the different network operating modes are the result of variations in symbol signaling rate. The bit rate governs the number of lanes contained in the frames propagated around the loop. The operating mode, and hence the bandwidth, of a loop is determined at initialization time and remains in effect until the next network reset. A network operating in MW25 mode would have 64 lanes per frame, one operating in MW50 mode would have 128 lanes per frame, etc.

In preferred embodiments of the invention, each frame begins with a frame header that occupies the first two quintets of the frame. Since each quintet occupies two lanes, the frame header occupies lanes 0-3. The "Command Stream" uses additional lanes 4-7. The Command Stream is the foundation of inter-device communication, and preferably has a permanent and invariant position in every frame. Other embodiments could use other lane assignments or different numbers of lanes for the frame header and Command Stream.

FIG. 9 illustrates one preferred embodiment of a frame header and command lane assignments. FIG. 9 shows the lane assignments for the frame header 902 and the Command Stream 904. All other lanes 906 in the frame 900 are available for media and data streams.

## Command Stream

In the most preferred embodiment, a Command Stream is used for resource discovery, resource allocation, configuration, and control. There are two levels of Command Stream protocols. The low-level protocols are used for direct control of devices. The high-level protocols, which are largely based on the Message command, support cooperation among  
5 processes that manage the network.

When describing devices, the terms "smart" and "dumb" will often be used. Smart devices cooperate with each other in the control of a network. This cooperation is based on a Command Stream and a common resource allocation model. All smart devices that manage streaming activities additionally implement certain higher-level resource allocation protocols.  
10 A smart device directs other devices by means of commands, while a dumb device does not. Generally, a smart device contains a microprocessor to support its application, and often includes a user interface. It optionally sends low-level configuration and control commands to other devices. It optionally hosts client processes that engage in transactions using a high-level protocol. A set top box and a pre-amplifier are examples of a smart device. A  
15 speaker is an example of a dumb device. A CD player may be a dumb device, even though it may include a microprocessor.

In the most preferred embodiment, commands are used to remotely read and write the memory and control registers of devices, and timed variants of some commands permit pre-scheduled, synchronized control of devices. A Command Stream supports basic discovery  
20 via a device scanning process, plus commands that allow remote access to Device ROMs (discussed in more detail below) in each device, that contain the device's self-describing data. The Clock Master, after establishing the Command Stream, broadcasts a single Scan Request command. Other devices may also request the Clock Master to perform a scan at any time. As each device sees the Scan Request, it in turn broadcasts a Scan Reply command. The reply  
25 includes the device's Globally Unique Identifier (GID), brief characterizations of the device's capabilities, and an address link to the self-describing data in the Device ROM. Because the replies are broadcast, any device may acquire complete basic information about the network configuration and the devices attached to it. When further information is wanted about a device, its Device ROM may be read remotely using a Read command.

### 30 Scan Reply

A Scan Reply command is sent as a delayed response to a Scan Request command. In a preferred embodiment of the invention, all devices are capable of broadcasting a Scan Reply command. Smart devices that control other devices preferably examine all Scan Reply

commands broadcast during network scan; the information contained in the reply is essential for network management. The information included in the Device Data field of the reply is primarily intended to help smart devices find or discover other devices without having to directly query each device. More detailed information is available in the self-describing data in the Device ROM.

In the most preferred embodiment, the bytes of the Device Data field are interpreted as shown in Table 2:

[0]	Protocol Version Number
[1-3]	Company ID
[4-6]	Model ID
[7-9]	Protocol Hint Bit Mask
[10-12]	Capability Hint Bit Mask
[13]	Max Streams - the number of streams implemented by the device
[14-16]	Start address of the device's Allocation Marker in RAM
[17-19]	Start address of the device's Device Information in ROM
[20-21]	Device Status
[22]	Loop Number
[23]	Loop Operating Mode

Table 2 – Device Data Fields

In preferred embodiments, the Company ID is a 3-byte IEEE Organizationally Unique Identifier, and the Model ID is a company-specified number. The Protocol Hint and Capability Hint bit masks are described below. The Device Allocation Marker is the address to be written on when claiming the right to operate or manage the device. The Device Information address is the address to be read in order to acquire the self-describing data

contained in the device's ROM. All of the above fields are static, except for the Device Status, Loop Number and Loop Operating Mode fields described below. The bits of the Protocol Hint Bit Mask are listed in Table 3 according to one preferred embodiment, but other bit assignments would be feasible in other embodiments.

5

0	Clock Master Protocol
1	Stream Control Protocol
2	Phase Lock Protocol
3	Time Stamped Phase Lock Protocol
4	Timed Device Control Protocol
5	Time Stamped Read Protocol
6	Peer-to-Peer Protocol
7	Switch Protocol

Table 3 – Bit Assignment in the Protocol Hint Bit Mask

The bits of the Capability Hint Bit-mask are defined in Table 4 according to one preferred embodiment, but other bit assignments would be feasible in other embodiments.

10

0	Reads Video
1	Writes Video
2	Reads Audio
3	Writes Audio
4	Uses MAPS
5	Uses Phone
6	Uses Misc.

16	Sensor
17	Effector
18	Player
19	Store
20	Transmitter
21	Receiver
22	Host
23	User Interface

Table 4 – Bit Assignment in the Capability Hint Bit Mask

In one preferred embodiment, the values of the Loop Operating Mode are assigned as listed in Table 5:

0	MW25
1	MW50
2	MW100
3	MW200

Table 5 – Bit Assignment for the Loop Operating Mode

Device Status is two bytes of arbitrary device status information. The interpretation of this field depends entirely on specifics of the device sending the Scan Reply command. One possible use of the first byte of the device status is to contain the current state of the GPIO lines.

## 10 Device Rom

In preferred embodiments, each device can convey information about its configuration and capabilities. This is accomplished in two stages. First, every device transmits hints indicating its basic function(s), which commands it supports, and other high level information. Second, every device supports a command allowing its memory to be examined over the network, and contains a Device ROM. A pointer to the start of the Device ROM is returned in the device's Scan Reply.

In general terms, the Device ROM is a hierarchical database containing attribute and value bindings. A Device ROM provides the ability to find the value of a given attribute of a device in a simple, flexible, and efficient manner. In one preferred embodiment, the Device ROM format is based loosely on the ANSI/IEEE Standard 1212 (1994 Edition). It is

5 comprised of two basic data structures: leaves and directories, both of which start with a common header format.

In one of many preferred embodiments, a header is a 32-bit value consisting of a 16-bit length followed by a 16-bit CRC-16 value, as shown in Table 6. The length specifies the size of the data that follows the header, and does not include the size of the header itself. The

10 CRC-16 field uses the polynomial specified for the Error Detect field and applies to the length field in the header and the data that follows the header.

Length (16 bits)	CRC-16
------------------	--------

Table 6 - Device ROM Header Format

### Leaf and Directory Formats

Leaves are single-valued objects consisting of a header followed by a stream of bytes.

15 A directory contains a header followed by directory entries. Directory entries contain immediate data, or point to leaves or other directories. The ROM format begins with the top-level directory. A directory entry consists of an 8-bit key followed by a 24-bit entry\_value, and one of many preferred embodiments is shown in Table 7.

Key_type	Key_value	Pointer or Immediate Value
2	6	24

Table 7 - Device ROM Directory Entry Format

### 20 Key Type Definitions

A key consists of a 2-bit key\_type followed by a 6-bit key\_value. The key-type indicates the type of the entry-value, and one preferred embodiment is listed in Table 8.

Reference name	key_type	Interpretation of entry_value
immediate	0	immediate value
not used	1	do not use

leaf	2	pointer in memo space to a leaf
directory	3	pointer in memory space to a directory

Table 8 - Device ROM Directory key\_type Definitions

For an immediate entry (key\_type = 0), the entry\_value is a 24-bit quantity. The meaning is dependent on the key\_value. For a leaf entry (key\_type = 2) or directory entry (key\_type = 3), the entry-value is the 24-bit address in the memory space of a leaf or directory object, respectively.

### Key Value Definitions

The key-value specifies the particular entry (e.g., Textual-Descriptor, Capability, etc.) as shown in Table 9, which gives some key\_value definitions used in the top-level directory according to one preferred embodiment, but other definitions could be used in other embodiments.

Entry Name	key_type(s)	key_value	Required
Textual_Descriptor	Leaf or Directory	01	
ROM_Format_Version	Immediate	02	Y
Version	Immediate	03	Y
Protocol_Hint	Immediate	04	Y
Capability_Hint	Immediate	05	Y
Max_Streams	Immediate	06	Y
Max_Write_Size	Immediate	07	Y
Start_Alloc_Marker_Address	Immediate	08	Y
Max_Operating_Mode	Immediate	09	Y
Company_ID	Immediate	0A	Y
Company_Name_Address	Leaf	0B	Y
Company_URL_Address	Leaf	0C	
Model_ID	Immediate	0D	Y
Model_Name_Address	Leaf	0E	Y
Serial_Number	Immediate or Leaf	0F	



BBOP_Bit_Array_Directory	Directory	10	
Streaming_Capabilities	Leaf	11	
IEEE_1212_ROM_Address	Immediate	12	
JINI_ROM_Address	Immediate	13	
XML_ROM_Address	Immediate	14	
Hardware_Devices_Directory	Directory	15	Y
MAC_and_PHY_Clock_Rates	Immediate	16	Y
Custom_Manufacturer_ROM	Directory	37	

Table 9 - Device ROM Top-Level Directory key\_value Definitions

**Logical Loop**

A network is a logical loop where frames continuously circulate around the loop visiting every device in the loop. The frame header immediately follows the last lane of the previous frame, so that every device sees frames arrive at exact periodic intervals. Devices index from the frame header to decide which lanes to read and which lanes to overwrite.

FIG. 10 illustrates one example of creating a logical loop 1000 from a daisy-chain. Although devices behave as if they were connected together in a loop architecture, the actual physical topology is a daisy-chain. In FIG. 10, the final link 1002 (dotted arrow) in loop 1000 is not actually connected. The loop 1000 is completed by the daisy-chain segments (black arrows) 1004, 1006, and 1008, which return all the streams from device D to device A as the dotted arrow would have done to logically complete loop 1000.

**Network Initialization**

When a network starts up, or when a device is added or removed from the network, an auto-negotiation procedure is performed. First, speed arbitration determines the network-operating mode. Clock arbitration then identifies which device will serve as Clock Master. When auto-negotiation is completed, the Clock Master device is selected and equal-sized frames are properly circulating around the loop. Auto-negotiation is a distributed process. No special controller is required to operate a loop according to a preferred embodiment of the invention. Auto-negotiation is discussed more fully below.

**Device Reset and Initialization**

FIG. 11 summarizes the top-level states and state transitions of a device in a network according to the most preferred embodiment. When the device is powered on, it and its neighboring devices immediately enter the Negotiating state 1102.

5 The Negotiating state 1102 represents the PHY-level auto-negotiation process in its entirety. During this process, the operating mode of the local loop is established and the Clock Master device is selected. All of the auto-negotiation device states are subsumed under Negotiating state 1102.

10 After the Clock Master device has been selected, it will 1) establish synchronized frame transmission, 2) initialize the token-based command transport mechanism, 3) start a network scan by broadcasting a Scan Request command, and 4) notify smart devices when network configuration is complete. Once synchronization has been established, the PHY level of each device generates a local Link Up notification and the devices move to the Passive state 1104. While in the Passive state 1104, devices neither stream onto the network, nor send commands to other devices.

15 Smart devices wait in the Passive state 1104 until the Clock Master has broadcast notification that the network configuration is complete. Following network configuration, smart devices enter the Active state 1106 and may begin to send commands to other devices attached to the network.

20 For streaming devices, the Passive state 1104 is implemented by a MAC level mode called Suspended Reset. In this mode all streaming is suspended. Smart streaming devices may leave the Suspended Reset mode under control of their local host as soon as network scan is complete. Dumb devices wait in the Suspended Reset mode until a Reset command specifying Exit Suspended Reset is received from whichever smart device is managing them. The dumb devices then move to the Active state 1106.

25 Before a process managing an activity on the network can restart streaming among the devices involved, it first revalidates the bandwidth allocation for the activity. If network bandwidth has decreased, the manager must reallocate bandwidth and reconfigure the streams used by the activity's devices. If network bandwidth has not decreased (and if no essential devices are unavailable due to network segmentation), the manager can usually restore the  
30 activity by broadcasting a single Reset command specifying the Exit Suspended Reset mode.

A second form of the Reset command, called the Begin Device Reset mode, is used to reinitialize the MAC level of a device without disturbing its PHY level or disrupting network

service. Following initialization of the MAC, the device is left in the Passive state (i.e., in the Suspended Reset mode). Smart devices relinquish their control of dumb devices by sending them a Reset command with Begin Device Reset mode. This form of reset is also appropriate for triggering via a reset button.

- 5           Hard reset of a device occurs only when the device is power cycled. This is because hard reset includes reset of the PHY level, which causes a temporary disruption in service in the local loop. PHY level reset brings down the links to the neighboring devices and forces a return to the Negotiating state 1102 to reestablish a network- operating mode.

#### **Clock Master Requirements**

- 10           Every network has exactly one enabled Clock Master (after the network completes the startup process). The Clock Master generates the network time base, and buffers the loop to an integral number of frame times in length. The Clock Master also initializes the clocks of all the devices on the network with the Time Mark command. Many devices (but not necessarily all) on the loop should be able to become the Clock Master. An auto-negotiation process  
15           arbitrates which of the eligible devices is selected to be Clock Master. The selected device remains the Clock Master until the next auto-negotiation.

- Devices that can source streams are able to become the Clock Master, unless the design of the application guarantees that another device that can perform this function will be present on the network. For example, telephones in a phone system designed to work only with a  
20           central switching controller need not be able to become Clock Master if the controller can perform the Clock Master function.

- The Arbitration Byte stored in the Device ROM specifies a device's priority in the clock arbitration scheme. This arrangement ensures that certain classes of device will always win over other classes. When two devices in the same class compete for Clock Master the  
25           decision is made arbitrarily by selecting the device with the higher GID.

            The Clock Master also performs some network initialization operations on the Command Stream and manages the Command Stream during normal operation. These responsibilities are described more fully in the Command Stream description.

#### **Network Time Coherence**

- 30           The MAC layer of each device keeps a clock based on a frame count. The clock rate in use in each device determines the precision with which time is measured in that device. The basic time base instability of the network is the result of the sum of the jitters of each of the

PHY chip phase-locked-loops (PLLs) between the Clock Master and a given device. An alternative embodiment measures data propagation times with a special timer resident in the PHY layers of both the device and the Clock Master that enables enhancing the absolute time accuracy of the network. The mechanism is described below.

5       The Clock Master in preferred embodiments initially synchronizes all the frame-counting clocks on the network by broadcasting a Time Mark command. This command specifies a frame count and is saved in a register in the MAC layer of each device that needs its frame count synchronized with other devices on the network. The Clock Master follows the Time Mark command with the transmission of a marked frame that goes around  
10   the network twice. The first time the marked frame is received after a Time Mark command, each MAC transfers the value saved from the Time Mark command into its own clock.

Devices with clocks synchronized in this manner have small errors due to the time it takes the marked frame to travel around the network. Each device's clock lags the Clock Master by the time it took the marked frame to reach the device. For many applications, this is  
15   sufficient accuracy because the delay will be less than 3.255 microseconds per device plus about one nanosecond for each foot of wire. Devices that require more accuracy than this standard clock synchronization mechanism provides may utilize an optional mechanism to enhance accuracy.

FIG. 12 illustrates an optional mechanism to enhance the accuracy beyond the  
20   standard clock synchronization. The PHY layer portion 1202 of this mechanism is implemented in all devices that are able to become Clock Master 1206 and all devices that need enhanced time accuracy. Each port of the PHY layer 1202 may observe the special marker as it passes in each direction. Each port, of the devices that support this mechanism, starts a timer at the PHY layer 1202 when it sends the marker and stops the counter when it  
25   next receives the marker. This counter in the PHY 1202 is at least 16 bits in length and its size is specified in the Device ROM of the device. A manager 1204 reads the timer values from all the devices that support this mechanism and calculates the network delays between them. It uses this information along with any internal delay specifications provided in the Device ROMs, to calculate offset values for the frame clock in each MAC layer 1208 of the devices.  
30   The MAC layer 1208 of each device supporting this increased accuracy has a register to store this offset information. This offset register is 4 bytes in length and represents the offset in units of  $1/65536$  of a frame. The manager then broadcasts a Time Mark Initiate command, which triggers the Clock Master to broadcast a Time Mark and follow it with a specially

marked frame to update all of the clocks in the network. On those devices supporting this optional mechanism, the MAC layer 1208, upon receiving the marked frame, adds the offset value from MAC offset register 1210 to the Time Mark value 1212 and uses that adjusted value as the new MAC clock value 1214. This mechanism synchronizes the clocks in devices  
5 that support it, limited only by the precision of the clocks in the PHY layer 1202 and MAC layer 1208, the accuracy of the internal delay specifications, and the calculations done by the controlling manager 1204.

In preferred embodiments, the MAC and PHY layers are clocked at different rates. This means that a manager that is using the optional accuracy enhancement method adjusts the  
10 values read from PHY layer counters before they are written to the MAC layer offset register. The PHY layer is clocked at a multiple of the symbol rate and the MAC layer is clocked at a rate that is a multiple of the quintet rate. The standard clock for the PHY is the MW200 mode symbol rate 61.44 MHz. The standard clock, at the MAC layer is four times the MW200 mode quintet rate, 49.152 MHz. Devices that require more precision may use higher clock  
15 rates. Such devices use clock rates that are 1, 2, 4, 8, 16, 32 or 64 times the standard rates. The resolution of time stamps used by commands ultimately limits their precision to a rate 64 times the standard MAC clock rate, corresponding to a time interval of 317.89 picoseconds. All devices have a Device ROM entry that labels both the MAC and PHY clock rates. The MAC layer uses a four-byte sub-frame counter, which is sufficient to handle any anticipated  
20 MAC layer clock rate. The standard PHY counter uses a two-byte counter. Any device utilizing a PHY clock rate greater than the standard PHY clock rate uses a three-byte counter. Alternative embodiments could use counters with a different number of bytes than the counters discussed above.

If the device is using the standard rates in both the PHY and MAC layers, then the  
25 manager using the PHY measured times to adjust MAC layer offsets multiplies the PHY measured count by 0.8 to account for the different clock rates. If the device is using non-standard clock rates, then the adjustment factor needs to be modified to reflect the clock rates in use.

### Performance Requirements

30 The performance of preferred embodiments of the invention is defined in terms of different parameters, as shown below.

Preferably, devices operate in the MW25 mode, and may operate in additional operating modes depending on the wire condition and the capabilities of the devices on the

network. Table 10 shows operating modes of networks with their associated symbol rates according to one preferred embodiment, but other embodiments of the invention could use other operating modes.

Name	Symbol Rate /Sec	Bit Rate	Lanes/ Frame	Overhead Lanes	Efficiency
MW25	7.68 M	12.288 Mbps	64	8	88%
MW50	15.36 M	24.576 Mbps	128	8	94%
MW100	30.72 M	49.152 Mbps	256	8	97%
MW200	61.44 M	98.304 Mbps	512	8	98%

Table 10 - Network Operating Modes

At each of the supported speeds and wire types the gross bit error rate in preferred embodiments is better than 10 bits in a trillion bits per link. This figure applies in the worst case of maximum operating mode and maximum cable length. This means that with five devices in a network, the system bit error rate will be better than 100 bits in a trillion bits (the complete logical loop passes through both links of each device, for a total of ten links).

The cumulative time base jitter from one device to any other in preferred embodiments is less than 10 nanoseconds measured over 100 milliseconds. Therefore, the jitter added by each device is preferably less than 0.1 ns with the total jitter accumulation of less than the system maximum over 100 devices.

#### 15 Network Initialization Time

The network initialization time includes the time it takes for a system to do all of the lowest level speed and clock arbitration and the higher level device scan and basic reset. Each device necessarily creates some delay in frame transmission around the loop, because of the time required to receive and regenerate the symbols (the cabling also creates delay, approximately 1 nanosecond per foot). In one embodiment, the minimum delay through a device is two quintets. That is, a device receives a whole quintet before re-transmitting it, since some line coding and scrambling algorithms have this requirement. Since the duration of a quintet is proportional to the operating mode of a network, the delay through a device is also proportional to the operating mode of the network. As the network runs faster, the delay through each device is reduced.

The required maximum propagation delay of a device that is not modifying a stream is specified as a function of the symbol time. In the most preferred embodiment of the invention, devices delay a non-modified stream by no more than the time taken by 25 symbols. For MW 100, which uses a symbol rate of 30.72 million symbols per second, the delay would be about 814 nanoseconds. The propagation through a device is less than the time given for each operating mode shown in Table 11, according to one preferred embodiment, but other device delay times would be feasible in other embodiments of the invention using other operating modes.

Operating Mode	Device Delay ns
MW25	3255
MW50	1628
MW 100	814
MW200	407

Table 11 - Device Delays Per Operating Mode

### Signaling

Information read from or written to the network is represented in different forms at different levels of the network architecture. FIG. 13 illustrates the most preferred embodiment for encoding symbols on the wire and at the PHY and MAC layers. The most preferred embodiments use a 5-level encoding scheme that provides redundancy in the data. This helps make the communications more robust, and permits the selection of a few unique quintet values 1302 that are used as control codes 1304. Values delivered to the MAC layer are presented as 4 bits and a flag for each lane. One possible set of values is shown in Table 12, but other embodiments of the invention could use other sets of values. As shown, the flag bit is set to zero when the 4 bits are a data value. If the flag bit is set to one, then the 4 bits represent a control code. Some control codes, such as the "RR" code, are not communicated to the MAC chip since the MAC chip is not involved in the startup process. 5B/4B data codes and control codes are discussed in detail in co-pending U. S. Patent Application Serial No. 09/079,914, entitled "Synchronous Network for Digital Media Streams," filed on May 15, 1998, which is incorporated by reference.

Code Type	Flag	Bit 3 2 1 0	Name	Description
Data	0	0 0 0 0	0	Data 0
Data	0	0 0 0 1	1	Data 1
Data	0	0 0 1 0	2	Data 2
Data	0	0 0 1 1	3	Data 3
Data	0	0 1 0 0	4	Data 4
Data	0	0 1 0 1	5	Data 5
Data	0	0 1 1 0	6	Data 6
Data	0	0 1 1 1	7	Data 7
Data	0	1 0 0 0	8	Data 8
Data	0	1 0 0 1	9	Data 9
Data	0	1 0 1 0	A	Data A
Data	0	1 0 1 1	B	Data B
Data	0	1 1 0 0	C	Data C
Data	0	1 1 0 1	D	Data D
Data	0	1 1 1 0	E	Data E
Data	0	1 1 1 1	F	Data F
Control	1	0 0 0 0	' I '	Idle
Control	1	0 0 1 1	' J '	Frame Marker, codes 1 and 2 of 4 ( 'JJ' )
Control	1	0 1 1 1	' K '	Frame Marker, codes 3 and 4 of 4 ( 'KK' )
Control	1	0 0 0 0	' R '	Auto-negotiation Message Marker ( 'RR' )
Control	1	0 0 0 1	' W '	Timing Frame Marker, replace J



Control	1	0 1 0 0	' T '	Token symbol
---------	---	---------	-------	--------------

Table 12 - Data and Control Codes at the MAC Level

### 5-Level Coding

In the most preferred embodiment of the invention, each device sends a series of 5-level symbols grouped into quintets. Each quintet contains five 5-level symbols. Therefore, a quintet is theoretically capable of representing one of 3,125 possible quintet values. Of these possible values, only 381 are DC balanced, or sum to zero. DC balancing reduces emissions because the level will sum to zero over most groups of quintets. 326 of these 381 balanced values are used for all the data quintets of a frame (i.e., all except the first two quintets in the frame). The first two quintets in each frame are limited to the 'JJKK', 'WWKK' and their inverses, all of which are DC balanced in pairs.

The DC balancing also leads to a method of doing forward error correction by exploiting the inherent redundancy. From the characteristics of DC balance, all data quintets must have at least two symbols that are different from any other data quintet. A received quintet that has one symbol received in error will not be DC balanced. The receiver may use a number of techniques to nominate the most likely balanced value and further qualify the quintet as one of the 326 that are actually used.

Table 13 shows the blocks of 5-level codes that are used to represent all the legal states possible in quintets encoded on networks, according to one preferred embodiment of the invention. Each quintet decodes into a code pair (i.e., two lanes worth of information).

Block Size	Most significant Code	Least significant Code
256 code pairs	0x0-0xF	0x0-0xF
16 code pairs	' I '	0x0-0xF
16 code pairs	0x0-0xF	' I '
16 code pairs	' T '	0x0-0xF
16 code pairs	0x0-0xF	' T '
1 code pair	' I '	' I '

1 code pair		'T'
1 code pair	'T'	'I'
1 code pair	'T'	'T'
1 code pair	'R'	'R'
1 code pair	RESERVED	
1 code pair	'J'	'J'
1 code pair	Inverted 'J'	Inverted 'J'
1 code pair	'K'	'K'
1 code pair	Inverted 'K'	Inverted 'K'
1 code pair	'W'	'W'
1 code pair	Inverted 'W'	Inverted 'W'

Table 13 - Network S-level Code Blocks

In the most preferred embodiment of the invention, the 'RR' code is only used during auto-negotiation. All variations of the codes 'J', 'W' and 'K' take up all five symbols to produce distinctive waveforms that span the frequency range and contain solid edges for synchronization. These codes are used for waveform references for equalization, echo cancellation, and polarity detection. They are unipolar, but they may be received in either polarity.

'JJKK' and 'WWKK' code pairs are used to mark the beginning of frames. Inverted frame markers are used for synchronizing the scrambling of codes. They consist of the inverted 'JJKK' and 'WWKK' codes. The inverted frame markers do not occur on two consecutive frames. Thus, the PHY layer hardware detects positive polarity by looking for two frame markers in a row with the same polarity.

The first 326 codes in Table 13, from the beginning through the reserved code, all exhibit zero DC offsets. FIG. 14(a) and 14(b) illustrate network 5-level data code pairs according to one embodiment of the invention. The first block of 256 codes is given values listed in FIG. 14(a) and FIG. 14(b). Each of the quintets in the first code block represent one 8-bit data value. The  $y[n]$  values are the symbol values before scrambling.  $Y[1]$  is transmitted first.

FIG. 15 illustrates 5-level control code pairs according to one embodiment of the invention. The remaining code blocks are given in FIG. 15. The quintets in these blocks contain one or more control codes and in the former case, may also contain a nibble of data.

### Scrambling

5       The most preferred embodiment of the invention transmits scrambled data signals for two purposes. Scrambling spreads out transmitted energy of the signal, reducing spectral lines (or peaks). Scrambling also reduces correlation in the data signal (and between signals on different cables), increasing the effectiveness of adaptive signal processing algorithms used in the receive circuitry. Preferred embodiments of the network interfaces utilize scrambling, as  
10 described in this section, during full-duplex operation. Half-duplex modes of operation during auto-negotiation do not use scrambled transmissions. Alternative embodiments could be implemented without scrambling.

### Side Stream Scrambler

      The most preferred embodiment utilizes a side stream scrambler. This technique  
15 combines pseudo random data with the actual data using a scrambling function prior to transmission. The pseudo random sequence is independent of the actual data. The receiver, generating the same pseudo random sequence as the transmitter, can then extract the original data stream by performing the inverse of the scrambling function. For example, if the transmitter has actual data 6 and a random number 2 then it adds the two values and transmits  
20 8. The receiver, using the same pseudo random sequence generator, also has random number 2 and the scrambled value 8. The receiver can subtract (inverse operation) 2 from 8 to get the original value of 6. Therefore, scrambling has 3 major components: pseudo random number generation, synchronization of transmit and receive pseudo random generators, and scrambling/de-scrambling functions.

### 25 Scrambler/De-scrambler Synchronization

      In order for the de-scrambler to recover the original data stream it must generate the same pseudo random sequence that the scrambler used to scramble the data stream. The pseudo random polynomial and initial value are determined during the early stages of auto-negotiation. Synchronization determines exactly when the initial value will be loaded.

30       De-scrambled frame markers are used to determine the exact point of synchronization. When a transmitter provides synchronization to the far end receiver it sends exactly one de-scrambled frame marker. The receiver always seeds the pseudo random number generator

immediately after the second quintet of a de-scrambled frame marker is received. Therefore, the receiver de-scrambles the quintet immediately following the de-scrambled frame marker using the pseudo random value. During full-duplex operation a de-scrambled frame marker only is sent to provide scrambler synchronization. Otherwise, frame markers will be  
5 scrambled using the functions described in the next section.

Scrambling of frame markers allows positive verification that the de-scrambler is still synchronized to the far end scrambler. A legal frame marker pair should always result from de-scrambling the quintets in the frame marker position. Otherwise, the scrambler and de-scrambler have somehow gotten out of sync. De-scrambled frame markers can be  
10 recognized immediately and are not de-scrambled. Reception of a de-scrambled frame marker always causes the de-scrambler to re-synchronize. The only likely cause of synchronization loss is a discontinuity in the frame index (i.e., a truncated or elongated frame). During normal operation this condition should never occur. If it does, the loop drops into auto-negotiation to re-configure the virtual loop.

15 When a de-scrambler gets out of sync with the far end scrambler, it requests re-synchronization. To do this the port with the out of sync de-scrambler sends a de-scrambled, inverted frame marker. Sending a de-scrambled frame marker (inverted or non-inverted) causes synchronization of the local scrambler and far end de-scrambler. When an inverted, de-scrambled frame marker is received, the receiving port synchronizes its  
20 transmit scrambler by sending a de-scrambled frame marker. Inverted frame markers are sent at most every other frame, and should occur much less frequently.

#### Scrambling / De-scrambling Functions

The line code imposes some constraints on the symbols that may be transmitted on the wire. In the most preferred embodiment, data is transmitted in groups of five symbols (i.e.,  
25 quintets) yielding 3125 possible combinations. To achieve several design objectives only 326 out of 3125 possible quintets are legal data quintets. Six quintets are defined for frame marker functions. To produce a legal 9B5Q data stream, the scrambled data quintets are in the legal set of 326 values. Therefore the scrambling/de-scrambling functions implement a one-to-one mapping from the legal set of 326 quintets to another legal set of 326 quintets.

30 In the most preferred embodiment there are 3 classes of quintets in scrambling: data, scrambled frame markers, and de-scrambled frame markers. Data is always scrambled in full-duplex operation. De-scrambled frame markers are used to synchronize the scramblers

and de-scramblers. Otherwise, frame markers are scrambled during full- duplex operation. The scrambler/de-scrambler functions process each quintet class differently.

During full-duplex operation, data quintets (i.e. non-frame marker quintets) are always scrambled. The scrambling function uses a pseudo random number to perform a rotation  
5 within the legal set of values (i.e., addition modulo 326). The de-scrambler inverts the process by performing the opposite rotation (i.e., subtraction modulo 326). A de-scrambled frame marker passes through the scrambler/de-scrambler unchanged. A frame marker is scrambled by subtracting 256 from the quintet value (i.e. to bring it within the legal data range), and the same function used for data quintets is performed. A de-scrambler, knowing the expected  
10 position of the frame marker, performs the de-scrambling function used for data and adds 256. A scrambler only produces a de-scrambled frame marker when requested by the synchronization circuit. A de-scrambler can recognize de-scrambled frame markers directly. A de-scrambler will notify the synchronization circuit when it receives a de-scrambled frame marker. A scrambler applies the frame marker scrambling function based on the input quintet  
15 value. A de-scrambler requires frame index information in order to apply the frame marker de-scrambling function based on position.

#### **Auto-negotiation**

Auto-negotiation has three purposes in preferred embodiments: determine the Clock Master, determine the operating mode, and configure the virtual loop. Once these three tasks  
20 are performed a device on the network can enter normal operation. In order to perform auto-negotiation, each device must be able to communicate to neighboring devices. To facilitate this communication, all devices preferably support half-duplex communication at the MW25 operating mode without scrambling. In the most preferred embodiment, auto-negotiation completes before the media access control protocols can be used (i.e.,  
25 command stream access, stream access). As a port proceeds through auto-negotiation it progresses through several states, communicating to its link partner via messages. At any given port certain conditions may persist that affect the auto-negotiation process.

#### **Major Auto-negotiation Phases**

As shown in FIG. 16, auto-negotiation is broken up into several major phases: link fail  
30 1602, clock arbitration 1604, speed arbitration 1606, loop configuration 1608 and normal operation 1610. The ports of a device proceed sequentially through these phases. At any point a device port may fall back to an earlier phase of the process as new information becomes available from other parts of the network or some disruption occurs (e.g. slow device

located elsewhere on network, link disconnected, etc.). Immediately following is an overview of each phase and its function. Following the overview, a detailed explanation states and actions is provided.

#### **Auto-negotiation Communication**

- 5           The full-duplex transceiver technology employed in a preferred embodiment of the invention requires each link to have a master and a slave where the slave recovers its timing from the master. In the most preferred embodiment, full-duplex communication does not take place until the Clock Master is determined. Therefore, at the start of auto-negotiation each device port only communicates with its nearest neighbor via half-duplex communication at the
- 10   MW25 operating mode. Once communication with the nearest neighbor is established, message passing from one port to another is used to communicate information across the whole network. When auto-negotiation is completed, the individual links will form a continuous virtual loop.

- Link partners exchange auto-negotiation information via messages, which are
- 15   embedded in a frame. One message is transmitted per frame. Any lanes in the frame that are not used by the message are forced to idle. Because of this and the fact the loop is not configured, applications do not attempt to transmit or receive data until the device has entered the normal mode of operation. The network hardware provides an indication to the application (hardware or software) whether or not the network is operating in normal mode.

- 20           The 9B5Q code is unipolar (i.e. the polarity matters). During half-duplex operation scrambling is disabled. Therefore, clearly identifiable frame markers appear at the beginning of every frame. Receivers search for the frame marker pattern and use it to establish the correct polarity and quintet alignment. The polarity determined during half-duplex operation is used during full-duplex operation (i.e., polarity does not change once full-duplex operation
- 25   has begun).

#### **Link Fail Phase**

- The link fail condition indicates that no communication is occurring. The most common reasons for a port to be in a link fail state are either that no cable is plugged in, or that there is no device on the other end of the cable. However, several other situations may cause
- 30   link fail. For example, the far end device may have lost power and stopped transmitting. The near end or far end device may have forced the link fail condition.

A device bypasses any ports in the link fail condition, because bypassing allows the virtual loop to remain complete. For instance, if a port on a device has no cable connected, the device passes the data to the next functioning port, allowing the stream to pass over unconnected or non-functioning ports. The link fail condition also has relevance to other stages of the auto-negotiation process. A device with all but one port in link fail is an endpoint, causing it to loop back the stream.

Two other conditions affect device port behavior in a similar way to link fail even though communications may occur over the link: link down and bad link. A device port may go into the link down state to indicate that a particular link should not be used during normal operation (i.e., to eliminate physical loops). A device port indicates a bad link if the quality of the physical medium creates an unacceptably high error rate. In these cases, the port is bypassed.

#### **Clock Arbitration Phase**

The most preferred embodiment of the invention requires that each link establish a master/slave relationship between the ports at either end of the link before any full-duplex communication may proceed. Therefore, clock arbitration is performed first using half-duplex communication to create a directed acyclic (non-cyclic) graph through all links of the network where each device is a node, and the clock master is the root node. Clock arbitration selects the Clock Master based on a 9-byte Arbitration Value. The most significant byte of the Arbitration Value is the Arbitration Byte. The Arbitration Byte prioritizes a device into one of 256 classes. A device's global identification number (GID) comprises the least significant 8 bytes its Arbitration Value. The device with the largest Arbitration Value is selected as Clock Master. The Arbitration Byte allows certain classes of devices to always win clock arbitration over devices of a lower class, and the GID breaks all ties. Not all devices are required to support the Clock Master function. An Arbitration Byte of 0 indicates the device is only capable of slave operation. If no master device resides on the network, auto-negotiation will not complete and the network will not enter normal operation.

To participate in clock arbitration a device continuously sends messages on ports. If the device receives an acknowledgment on all ports, then it has been elected the Clock Master. However, if it receives a message containing an Arbitration Value larger than its own then it has lost arbitration. If a device loses clock arbitration it will proxy the Arbitration Value of the winning device (i.e. send messages containing winner's Arbitration Value) out all other ports. If an acknowledgement is received on all of the other ports then it will send an

acknowledgment back to the winning device (i.e. on the same port that the winning message arrived). However, if a message with an even higher Arbitration Value arrives on any of the ports the device will proxy the new Arbitration Value out all of the other ports. This algorithm guarantees that the correct Clock Master has been selected when that device receives  
5 acknowledgements on all ports. However, new devices may be added to the network at any time. Therefore, clock arbitration may restart at any time.

### Half-Duplex Operation

During the clock arbitration and speed arbitration phases communications occur in a half-duplex mode. Half-duplex communications occur at the MW25 operating mode, and  
10 scrambling is disabled. Auto-negotiation begins in half-duplex mode for several reasons. Full-duplex communication requires that each link have one master and one slave. Until clock arbitration has created a directed ordering through all devices on the network, some devices may have two slave ports, and this creates a problem for full-duplex operation. Also, scrambler seeds must be exchanged between devices before full-duplex operation can proceed.  
15 Therefore, information must first be exchanged through some basic mechanism, which is the purpose of half-duplex operation.

FIG. 17 shows the state machine for half-duplex communication for one preferred embodiment. A combination of carrier sense and random wait is used to establish an alternating pattern of transmission between two link partners. The cycle begins with a random  
20 wait state 1702. Once entering the random wait state a device waits a random amount of time between 1 and 64 microseconds before proceeding to the transmit state 1704. The technique to generate random intervals is described below. If "carrier sense" is detected on the wire while the port is waiting then the device proceeds to the receive state 1708. If no energy is detected before the random delay expires, the port then transmits a preamble followed by at  
25 least four and no more than five complete frames with the pending transmit message in each one. The transmission need not begin on lane 0 of a frame. However, four complete frames must be transmitted. The preamble consists of at least 20 and no more than 40 alternating +2, -2 symbols.

After transmitting four complete frames, the port waits 350 nanoseconds to 450  
30 nanoseconds for the echoes to cease in state 1706. Then it begins listening for a link partner to transmit. If carrier sense is not detected within 100 microseconds, then the port begins the cycle again with a random wait 1702. If the port detects carrier sense, it attempts to receive



the signal until carrier sense subsides. Then it begins the cycle again by moving to the random wait state 1702.

### Audio Stream Format

Many different digital audio standards are appearing that address an ever-increasing variety of emerging applications. While the CD audio format has been successful for quite some time, new technologies such as multi-channel home theater systems and DVD audio are improving on the old standard. These new systems demand varied sample sizes and sample rates along with increased data rates. In order to handle all manner of audio streams, ranging from audiophile quality multi-channel formats to low bandwidth mono phone service, a flexible and comprehensive Audio Stream Format is used by the most preferred embodiments of the invention. The Audio Stream Format is designed to transport a vast variety of digital audio data formats, covering a large range of different sample sizes, numbers of channels, sampling frequencies, and encryption, compression and coding schemes.

Being synchronous, preferred embodiments transport audio streams with a fixed and stable time-base that supports very high quality audio. With no sample rate conversion or time-stamping, the Audio Stream Format makes it possible to interconnect audio equipment running at arbitrary sampling rates with the same high fidelity. In some cases, compressed audio that arrives in a burst of data rather than in a continuous flow can also be transported.

In the most preferred embodiment of the invention, the Audio Stream Format accommodates a wide range of sample structures, sample sizes and sample rates. This flexibility is the basis for supporting an open-ended variety of audio formats. Table 14 contains some examples of audio formats that can be handled using the Audio Stream Format.

Audio Format	Sample Size (bits)	Number of Channels	Sample Rate (kHz)	Bit Rate (kbps)		
				Min	Nominal	Max
Phone	8	1	8		64	
ISD	8	2	8		128	
Dolby				32	384	640
Dolby Digital (AC3) Compressed						
Dolby Digital	20	6	48		5800	

(AC3) Uncompressed						
CD Audio	16	2	44.1		1400	
DAT	16	2	48		1500	
24 Bit Audio	24	2	48		2300	
DVD Audio				1400		9600

Table 14 - Compatible Audio Formats

The sequencing of audio samples within structured audio streams is standardized in such a way that high fidelity formats can be directly decoded by low fidelity devices with no need for re-routing or mixing. This conversion capability is a key advantage of the Audio Stream Format. It lowers system complexity, reduces parts counts, and grants enormous flexibility to product designers and end-users alike.

The Audio Stream Format specification extends the structure of Device ROM so that processes managing audio activities can query the capabilities of audio components via the Command Stream in order to determine what audio formats they support. Whenever an application must send audio from a source device to a consumer device, the application manager reads both devices' Device ROM to determine their audio streaming capabilities. It then selects the audio format that best suits the combination of the media, producer and consumer.

Streams encoded in the Audio Stream Format may contain embedded audio commands that support advanced features or confer greater flexibility in handling the stream. They may be used to provide any extra information needed for retrieving, decoding, processing, storing or editing audio streams. Among other things, in-stream commands can be used to support copy-protection and encryption.

A special audio streaming mode allows a limited number of devices to mix independent audio sources into a single stream. This is particularly useful for telephony applications.

### Terminology

The term "audio stream" refers to a flow of audio data moving on a network and encoded using the Audio Stream Format. An audio stream consists of a sequence of audio packets that succeed each other at the audio sampling rate of the stream.

There are three different types of audio streams: frame synchronous, clock synchronous, and asynchronous audio streams. An audio stream running at the frame rate (48 kHz) is the prime example of a frame synchronous stream. In such a stream, each frame contains one audio packet, and the respective nibbles of each packet are mapped onto the same lane in every frame. Audio streams running at binary multiples of the frame rate are also considered to be frame synchronous. In these streams, which are termed "over-sampled frame synchronous," each frame contains several audio packets. The packets and their constituent nibbles repeat in exactly the same lane positions from frame to frame.

Finally, frame synchronous streams may run at binary sub-multiples of the frame rate. In these cases it takes some fixed number of frames to deliver each packet. Such streams are said to be "sub-sampled frame synchronous."

An audio stream whose packets permute through a repeating pattern of nibble-to-lane mappings over some fixed number of successive frames is referred to as a "clock synchronous" audio stream. Such a stream requires more complex decoding than a frame synchronous stream does, but its time-base is actually tied directly to the clock rate. For example, 44.1 kHz can be derived from the clock through the exact ratio 147/160.

An audio stream running at a sampling rate extracted from an external clock having no relationship to the frame rate is referred to as an "asynchronous" audio stream. Although CD audio can be encoded as a clock synchronous stream, in practice it is treated as an asynchronous stream.

Each audio packet is composed of one or more audio data samples (depending on the number of audio channels being conveyed by the stream). An 'Idle' code or in-stream command may mark the end of the packet, depending on the type of the audio stream. The terms "audio stream sample" or "stream sample" refer to the audio content of an audio packet (i.e., it is considered apart from its encoding).

#### **Stream Definition**

Audio streams are composed of audio packets. Each audio packet carries a single stream sample, which in turn is comprised of at least 1 and not more than 32 audio samples. The minimum size of a stream sample is 2 nibbles, and the maximum size is 64 nibbles.

For frame synchronous audio streams, each audio packet is explicitly terminated or delimited. For clock synchronous and asynchronous streams, at least every other audio packet is explicitly terminated. In all cases, the delimiter is either a single "idle" control code or an

in-stream command. In-stream commands consist of a "T" control code followed by an extension consisting of a single data nibble.

- For any given audio stream, each frame may carry anywhere from a fraction of a single audio packet on up to several complete audio packets, all as part of the same stream. The network bandwidth required to carry the stream may vary from a minimum of one lane per frame up to the maximum allowed by the network-operating mode.

### Supported Stream Types

Some audio stream types, audio sample sizes, and stream sample rates that are supported by preferred embodiments of the invention include:

#### 10 Frame Synchronous Streams

Audio Sample size	8, 12, 16, 20, 24 and 32 bits.
Stream Sample Rate	48 kHz, or binary & sub-multiples of 48 kHz

#### Clock Synchronous Streams

Audio Sample size	8, 12, 16, 20, 24 and 32 bits.
15 Stream Sample Rate	derived from the sampling rate (includes 48 kHz)

#### Asynchronous Streams

Audio Sample size	8, 12, 16, 20, 24 and 32 bits.
Stream Sample Rate	8 to 48 kHz or greater with multi-sampling

- The encoding of frame synchronous streams is optimized to take advantage of the special regularities of this type of stream. Clock synchronous and asynchronous audio streams are encoded using the same basic scheme. The only difference in treatment between the two is that their time-bases have different sources.

- In a preferred audio stream, each stream sample includes at least one audio sample, and each audio sample is at least 2 nibbles in size. Preferably, all audio samples are encoded most significant bit first. The stream samples of a multi-channel audio stream each contain the same number of audio samples for each channel (all channels of a multi-channel stream are encoded at the same sampling frequency). All audio samples within the stream sample share the same format and are of the same size.

- Preferably, stream samples do not exceed 64 nibbles in size. Since each audio sample is at least 2 nibbles in size, and since all audio samples within the same stream are the same

size, it follows that the stream samples of an audio stream may consist of up to 32 audio samples each. This provides the basis for a wide variety of multi-channel and/or multi-band formats. The following stream sample formats are supported for frame synchronous, clock synchronous, or asynchronous audio streams:

- 5            Mono-channel    Single channel audio stream (e.g., telephone, intercom, etc.).
- Multi-channel    Two or more time coherent audio channels combined in a single audio stream (e.g., stereo channel, Dolby Digital, Dolby Surround)
- Multi-process    Two or more audio channels derived from processing of a single channel (e.g., mufti-band equalizer output, cross-over output)
- 10           Multi-sampled    In a mufti-sampled stream, a stream sample contains several audio samples of the same channel. The audio samples run at a multiple of the stream sample rate.
- Mufti-source       Two or more audio sources mix onto a single full-duplex audio stream. When the audio stream arrives at a device stream, the
- 15           previous sample sent by this source is subtracted, and the next sample is added in before the stream is re-injected on the network

#### Audio Sample Precedence

In one preferred embodiment of multi-channel audio streams, the audio samples contained in each stream are ordered in the following precedence:

- 20           1.    by channel number in ascending order or (Left, Right)
- 2.    by channel type (e.g. in pass-band order: high band, middle band, low band)
- 3.    by sample number (for mufti-sampled streams)

For example, in a stereo twice-multi-sampled 2-band audio stream, audio samples will appear in the stream sample in the following order:

- 25    [LeftHigh1, RightHigh1, LeftLow1, RightLow1, LeftHigh2, RightHigh2, LeftLow2, RightLow2]

Channel Number	L	R	L	R	L	R	L	R
Channel Type	H		L		H		L	
Sample Number	1				2			

Table 15 - Audio Sample Ordering in a Multi-Channel Audio Stream Example

Each audio packet in a frame synchronous stream is explicitly terminated with an end-of-packet delimiter, either a single "Idle" code or an in-stream command. FIG. 18 illustrates one example of coding of a 20-bit sample 1802 on an audio stream, and shows how a stream sample consisting of a single 20-bit audio sample is formatted and distributed onto lanes 1804 in a frame 1806 beginning with frame header 1808. Exactly which lane positions in the frame 1806 will be utilized is determined when the stream is initialized and lanes 1804 are assigned to carry it. The audio sample is cut into individual nibbles. Each successive nibble is coded appropriately and transmitted in one of the lanes 1804 assigned to the stream. Finally, the delimiter 1810 (in this case an "Idle" control code) is inserted to mark the end of the audio packet.

Several samples can be sent in each stream sample in order to create a multi-channel stream. FIG. 19 shows how a stream sample consisting of a stereo pair of 20 bit audio samples 1902 and 1904 would be formatted and distributed onto the frame 1906 with frame header 1908, lanes 1910, and packet delimiter 1912. This stream sample could be seen as composed of two 20-bit samples, or as a single 40-bit sample. The encoding and decoding of audio streams must be externally coordinated using information that is not directly contained in the stream.

## 20 Media Asynchronous Packet Stream Format

In the most preferred embodiment, a Media Asynchronous Packet Stream (MAPS) is used by network protocol clients (e.g., TCP/IP) to send data packets over networks. MAPS streams are shared among peer reader/writer devices using an access method derived from the IEEE 802.5 standard for Token Rings.

A number of terms used to describe MAPS are applied in a manner consistent with IEEE LAN standards and may conflict with usage elsewhere in this application. This glossary is included to clarify the use of the following terms within the MAPS context.

**frame:** A sequence of bytes transmitted over a set of lanes allocated to MAPS, delineated by a unique non-data sequence called the Starting Delimiter and a unique non-data sequence called the Ending Delimiter. In any case where the meaning may be ambiguous, the term frame is used to distinguish a frame from a MAPS frame.

LLC: Logical Link Control layer, the upper part of the data link layer in an OSI architecture.

MAC: Medium Access Control layer, the lower part of the data link layer in an OSI architecture. In any case where the meaning may be ambiguous, the term MAC is  
5 used to distinguish the MAC from the MAPS MAC.

ring: A group of stations connected by lanes and using the MAPS protocol to share access to the lanes. A ring should not be confused with a loop.

station: A device capable of performing the MAPS functions.

In this description, MAPS is represented as collections of octets (bytes). MAPS  
10 operates on a byte stream, therefore the transmission order of frame headers and data is resolved to the byte level with bytes being transmitted in most-to-least significant (left-to-right) order. Each byte is composed of bits numbered 7-0, where bit 7 is always the most significant bit (msb) and bit 0 is always the least significant bit (lsb) in memory. In some cases, non-data codes are shown or described as nibbles for simplicity. Strictly speaking,  
15 these codes and data nibbles each occupy a lane on the physical medium.

The MAPS access method is based on the classic Token Ring access protocol (TKP) developed by IBM and later standardized in the United States by IEEE and internationally by ISO/IEC. However, the MAPS frame format, maximum frame size, and bit order of LAN addresses in memory have been adopted from IEEE 802.3 CSMA/CD (Ethernet) in order to  
20 simplify bridging to this popular LAN standard.

The TKP standard was designed to detect and isolate many common LAN problems such as faults in the physical wiring plant or network interface. TKP also provides for different priorities of service, enabling low latency application streams. These functions are handled by the architecture of the most preferred embodiment and do not need to be duplicated  
25 in MAPS. Therefore, MAPS provides a subset of TKP functionality limited to best effort delivery of single priority data frames. MAPS is a contention-free communication technique in which a group of stations are (logically) connected in a ring and a token (a three-byte entity) is used to control ring access.

FIG. 20 depicts four stations on a ring and illustrates the four basic steps of MAPS  
30 operation; token circulation 2002, transmitting 2004, receiving 2006, and stripping 2008. Each station receives data from its upstream neighbor and transmits data to its downstream neighbor. The token, which is initially released by one station on the ring termed the Active

Monitor (AM), circulates from station to station around the ring with each station repeating the data (2002). A station wishing to transmit data to another station must wait for a token to be received and, after detecting the start of the token, it changes the token to a start-of-frame sequence and appends the frame fields (Station B in 2004). The MAPS frame circulates  
5 around the ring, repeated by each station. Any station that recognizes the destination address of the frame (Station D in 2006) copies the frame to a local buffer in addition to continuing to repeat the frame to the next station on the ring. Finally, the originating station strips or removes the frame from the ring and releases a token (Station B in 2008).

### Conclusion

10 A logical ring network can run over the existing analog telephone wire segments in a home. Moreover, in a preferred embodiment of the invention, virtually any physical topology (star, loop, tree, and so forth) throughout a home, office, or other environment can be converted, with minor junction-box wiring modifications, into a logical ring network. Information can propagate around this logical ring, reaching every device on each revolution  
15 around the network. Network devices can be full-duplex, transmitting, and receiving information simultaneously.

An arbitration process occurs automatically upon network initialization, and one of the competing devices is elected the network clock device to which all other devices are then synchronized. By synchronizing all network devices to a single reference clock, and providing  
20 fixed frames of information propagating around the network at consistent time intervals (e.g., 48 kHz "frame rates" in one embodiment), the logical ring network ensures that information always will propagate from one device to another at consistent time intervals. In other words, information will propagate consistently around the logical ring network at the frame rate; and the time required for information to propagate between any two particular devices will remain  
25 fixed.

Following network initialization (and whenever devices are added or removed), an auto-configuration process configures each network device and determines the network topology. The entire network topology is discerned and made available to interested devices, even though individual devices need not be capable of interpreting such information. Devices  
30 can be added and removed in true "plug and play" fashion, and will be hot-pluggable if connected anywhere on a chain of devices attached to a "hot-pluggable smart jack" device.

Most network devices are relatively "dumb" and inexpensive, in that they require only simple hardware state machines to accommodate the basic network protocol, and to transmit



and/or receive digital media streams. Network devices have unique static device IDs, which simplify device identification and addressing, as well as network initialization, and form the basis for encryption to provide network security, authentication and/or copy protection functionality. Devices also can contain other device-specific information, including device  
5 drivers (or pointers to external device drivers) that can be executed on their behalf by "smart" CPU-based controllers.

In addition to accommodating adapters that connect existing consumer electronics devices to a logical ring network of a preferred embodiment of the invention, the network architecture accommodates new "restructured" digital-ready devices that redistribute existing  
10 device functionality across the network. For example, by removing MPEG2 decoders from DVD players, the compressed digital information can be distributed/processed throughout the logical ring network before reaching its ultimate destination (e.g., a television attached to or incorporating an MPEG2 decoder).

Information propagates along the logical ring network in fixed-length frames. In  
15 addition to "frame header" markers that synchronize devices to a master clock, these frames contain two independent streams: a "data stream" for the distribution of real-time continuous digital media streams, as well as asynchronous data, and a "system command stream" for the distribution of "system commands," which are used primarily for network initialization and auto-configuration of network devices, as well as basic switching of digital media streams.  
20 The system command stream propagates along a "default network path" that reaches every network device. The data stream, however, can propagate along any available path, to provide for greater overall network bandwidth (e.g., by adding a switching router device to the network to create alternative data stream paths).

The data stream is divided into distinct "channels" (the size of each channel being  
25 tailored to the bandwidth and sample-rate requirements of a particular media type) that operate by default, and can be reallocated dynamically (in some cases even occupying noncontiguous portions of the data stream). In this manner, devices can reliably guarantee consistent delivery of data (e.g., audio samples at the standard CD audio 44.1 kHz rate) having particular bandwidth requirements. Thus, two speakers on the network will receive left and right  
30 channel digital audio, respectively, at the "same time" (i.e., within one sample time of accuracy), and thus be synchronized to each other (i.e., phase coherent), even if the source device is physically located in another room and/or zone of the network). Phase coherency is

a critical factor in high-quality stereo audio systems, as well as multi-channel surround sound systems (e.g., Dolby AC-3).

The data stream also contains embedded control information and other asynchronous data, including, compressed MPEG2 video and other variable bit-rate data, as well as  
5 asynchronous network protocols, such as VC, RS232 serial protocols, and TCP/IP. Such information is delivered synchronously, thereby avoiding collisions. Network devices can utilize channels of the data stream to send custom commands to one another, including control information, pursuant to any protocol known to such devices.

A preferred embodiment of the invention is a synchronous, peer-to-peer, point-to-point  
10 network, connects all devices in one serial loop via a single pair of wires. For best performance and longest distance between devices, the wire should be category 5 grade twisted pair. However, the invention can operate at lower speed over lower quality wire, including untwisted pair or even the flat ribbon wire used to connect telephones to wall jacks. Network speed is limited by the maximum signaling speed of the attached devices. When the  
15 network is started, communication along each link is validated at the lowest speed (25 Mbps), but if every link can support it, the network speed, is increased.

In whole-house installations, the existing telephone wiring may often be used to create a home network. Ordinary Category 3 or Category 5 telephone wire is used to connect devices. The preferred embodiment also tolerates the lower-quality telephone wiring found in  
20 many homes including unshielded twisted-pair 'Bell' wiring, even unshielded untwisted 'Quad' wire. When run over poor-quality wiring, the network interface detects bad wiring conditions and automatically remains at a slower speed (25 or 50 Mbps total throughput) to compensate.

Preferred embodiments of the invention can connect devices over long distances.  
25 Using Category 5 telephone wiring, individual devices can be up to 100 meters (over 300 feet) apart. If Category 3 wire is used, devices should be no more than 50 meters (about 150 feet) apart. In any case, preferred embodiments of the invention can support at least 100 devices connected by thousands of meters of wire. Preferred embodiments of invention support many types of devices and carry many types of media and data:

- 30
- Audio: Nearly any type of digital audio including AC-3 (digital surround sound) and new high sample rate formats such as DVD audio;

- Video: MPEG1 and MPEG2 video, including DVD video and HDTV (high-definition television) signals which can require up to 19.2 Mbps);
- Voice telephony, FAX signals, or ISDN lines which are encoded as low data rate digital audio;
- 5      • TCP/IP (i.e., Internet) data; and
- Other standard data formats including RS-232, MIDI, CE IR, IrDA, and X-10.

One preferred embodiment of the invention achieves data rates of up to 98 Mbps. Even assuming the worst-case scenario of half-duplex broadcast of all information to all points on the network, that's enough bandwidth to simultaneously carry: eight high-quality MPEG2  
10 video channels (6 Mbps each) with accompanying audio; thirty-two 24-bit audio channels; sixteen phone or ISDN lines (encoded as low bit-rate digital audio); and more than 8 Mbps of asynchronous data such as serial or TCP/IP data.

Preferred embodiments of the invention are extremely bandwidth efficient. Up to 98% of network bandwidth is reserved for end-user media and data. Because preferred  
15 embodiments of the invention use a synchronous architecture, media need not include time stamps, which consume half of available bandwidth in some prior art networks.

Because preferred embodiments do not rely on a packet-based architecture, devices on the network are not required to have processors. To include devices without processors, the preferred embodiment of the MAC level directly implements a distributed architecture that  
20 allows network applications to be controlled from a small number of devices containing processors.

The preferred embodiments of the invention provide elegant and easy-to-install systems with a wide variety of applications including:

- Whole-house media distribution and networking;
- 25      • Home theater interconnection;
- Audio component interconnection;
- Home control and automation;
- Home PC networking;
- In-vehicle media distribution;
- 30      • Multiple line PBX phone systems.

The exemplary embodiments described herein are for purposes of illustration and are not intended to be limiting. Therefore, those skilled in the art will recognize that other embodiments could be practiced without departing from the scope and spirit of the claims set forth below.

**What is claimed is:**

1. A method for communicating information by using a plurality of symbols generated by a source device on a network for transmission in said network by said source device and  
5 reception in said network by a destination device, said method comprising:  
    encoding said information to produce an encoded symbol;  
    expressing said encoded symbol as a scrambled multi-level electrical signal at said source device, wherein said multi-level electrical signal has at least three levels;  
    de-scrambling said scrambled multi-level electrical signal at said destination device to  
10 determine said encoded symbol;  
    grouping at least one of said encoded symbols into a symbol group; and  
    decoding said symbol group into said information.
2. The method of claim 1, wherein said multi-level electrical signal has at least five levels.
- 15 3. The method of claim 1, wherein said symbol group is comprised of at least five symbols.
4. The method of claim 1, wherein said symbol group represents a plurality of binary bits.
5. The method of claim 1, wherein said symbol group represents nine binary bits.
6. The method of claim 5, wherein said nine binary bits can selectively represent a first  
20 plurality of data values or a second plurality of non-data values.
7. The method of claim 1, wherein said symbol group is substantially DC balanced.
8. The method of claim 1, wherein said symbol group is transmitted from said source device to said destination device on a connection selected from the following group, including a single pair of wires, a double pair of wires, a coaxial cable, one optical fiber, and two optical  
25 fibers.
9. The method of claim 1, wherein said symbol group represents at least two lanes of information.

10. The method of claim 9, wherein a stream from said source device is carried by said at least two lanes of information representing data and control codes.
11. The method of claim 9, wherein a data frame comprising said at least two lanes is propagated from said source device to said destination device.
- 5 12. The method of claim 11, wherein a data frame marker is detected in said scrambled multi-level electrical signal at said destination device.
13. The method of claim 12, wherein said data frame comprises at least two streams of data and an error in one stream of data in said data frame does not reduce the integrity of the remainder of said streams of data in said data frame.
- 10 14. The method of claim 13, wherein said stream of data in at least one stream packet is selectively overwritten without processing a whole data frame of said data at said destination device.
15. The method of claim 1, wherein said symbol group contains a plurality of evenly spaced codes with a minimum spanning distance of two between said codes to facilitate error  
15 detection and correction at said destination device.
16. The method of claim 15, wherein the data in at least one stream packet is processed for error detection and error correction at said destination device.
17. The method of claim 16, wherein error correction is performed at said destination device using computation of all possible single distance errors.
- 20 18. The method of claim 1, wherein the total delay time for information decoding and information re-encoding of a stream at said destination device is more than or equal to the transmission time for ten symbols from said source device to said destination device.
19. The method of claim 1, wherein transmission of said scrambled multi-level electrical signal from said source device to said destination device generates substantially a broad band  
25 pure noise.
20. The method of claim 1, wherein a sequence of decode logic and a lookup table is used for decoding at least one of said symbol groups into said information.

21. A method for transmitting a command stream generated by a source device for reception by a destination device in a network connecting a plurality of devices, said method comprising:
- appointing one of said plurality of devices as a clock master to provide a command stream token on said network, wherein each said source device on said network is required to receive said command stream token before transmission of said command stream, and hold said command stream token until transmission of said command stream is complete;
  - releasing said command stream token to another device on said network;
  - encoding said command stream as a group of one or more symbols, wherein each symbol is represented by a multi-level electrical signal having at least three levels;
  - receiving at said destination device one or more said multi-level electrical signals representing said group of one or more symbols; and
  - decoding said group of one or more symbols to reconstruct said command stream at said destination device.
22. The method of claim 21, wherein said step of appointing one of said plurality of devices as said clock master occurs at a start-up of said network, or when one or more devices are added or removed from said network.
23. The method of claim 21, further comprising:
- performance testing said plurality of devices at a start-up of said network using a default transmission rate from a plurality of transmission rates; and
  - selecting a transmission rate from said plurality of transmission rates that is supported by said plurality of devices.
24. The method of claim 21, further comprising the step of configuring a loop in said network with substantially continuous back-to-back sequences of frames comprised of said command stream and one or more additional streams.
25. The method of claim 21, wherein said multi-level electrical signal has at least five voltage levels.
26. The method of claim 21, wherein said group of one or more symbols is comprised of at least five symbols.
27. The method of claim 21, wherein said group of one or more symbols represents a plurality of lanes of information.

28. The method of claim 21, wherein at least two of said group of one or more symbols represent said command stream.
29. The method of claim 21, wherein said command stream is comprised of a variable number of groups of one or more symbols.
- 5 30. A method for transmitting an audio stream generated by a source device on a network for reception in said network by a destination device, said method comprising:
- (a) expressing said audio stream as a group of one or more symbols, wherein each symbol is represented by a scrambled multi-level electrical signal having at least three levels;
  - (b) de-scrambling and error correcting said scrambled multi-level electrical signal
  - 10 at said destination device to determine said group of one or more symbols;
  - (c) grouping said one or more symbols into a symbol group to be decoded; and
  - (d) decoding said symbol group to reconstruct said audio stream at said destination device from at least two of said symbol groups.
- 15 31. The method of claim 30, wherein said multi-level electrical signal has five voltage levels.
32. The method of claim 30, wherein said symbol group is comprised of at least five symbols.
33. The method of claim 30, wherein said symbol group represents a plurality of lanes of audio information.
- 20 34. The method of claim 33, further comprising:
- appointing one of a plurality of devices connected to said network as a clock master at start-up of said network, or when one or more devices are added or removed from said network; and
  - selecting a transmission mode from a plurality of transmission modes that is supported
  - 25 by a device of said plurality of devices and that is supported by a link connecting said device to said network.
35. The method of claim 30, wherein said audio stream is transmitted from said source device to said destination device at a substantially constant frame rate.



36. The method of claim 35, wherein said audio stream is frame synchronous, wherein one audio packet is contained in a network frame and the respective nibbles of each audio packet are mapped onto the same lane in every said network frame.
37. The method of claim 35, wherein said audio stream is clock synchronous.
- 5 38. The method of claim 35, wherein said audio stream is running asynchronously at a sampling rate extracted from an external clock independent of said network frame rate.
39. A method for transmitting an asynchronous packet stream generated by a source device on a network for reception in said network by a destination device, said method comprising:
- 10 (a) expressing said asynchronous packet stream as a group of one or more symbols, wherein each symbol is represented by a scrambled multi-level electrical signal having at least three levels;
- (b) de-scrambling said scrambled multi-level electrical signal at said destination device to determine said group of one or more symbols;
- (c) grouping said one or more symbols into a symbol group to be decoded; and
- 15 (d) decoding said symbol group to reconstruct said asynchronous packet stream at said destination device from at least two of said symbol groups.
40. The method of claim 39, wherein said multi-level electrical signal has five voltage levels.
41. The method of claim 39, wherein said symbol group is comprised of at least five
- 20 symbols.
42. The method of claim 39, wherein said symbol group represents a plurality of lanes of information.
43. The method of claim 39, wherein said asynchronous packet stream is comprised of a variable number of symbol groups.
- 25 44. The method of claim 42, wherein said asynchronous packet stream contains a plurality of evenly spaced codes with a minimum distance of two between said codes to facilitate error detection and correction at said destination device.
45. The method of claim 42, wherein the data in at least one asynchronous packet stream packet is processed for error detection and error correction at said destination device.

46. The method of claim 45, wherein said error correction uses all possible single distance errors in said asynchronous packet stream.
47. The method of claim 42, wherein the total delay time for data decoding and data re-encoding of an asynchronous packet stream is more than or equal to the transmission time for  
5 ten symbols from said source device to said destination device.
48. The method of claim 42, wherein transmission of said asynchronous packet stream from said source device to said destination device also generates noise that is substantially a broad band pure noise.
49. The method of claim 42, further comprising:  
10 appointing one of a plurality of devices connected to said network as a clock master at start-up of said network, or when one or more devices are added or removed from said network.
50. The method of claim 42, further comprising:  
appointing one of a plurality of devices connected to said network as a clock master;  
15 selecting a transmission mode from a plurality of transmission modes that is supported by a device of said plurality of devices and that is supported by a link connecting said device to said network; and  
configuring a loop in said network with substantially continuous back-to-back sequences of frames comprised of said plurality of lanes.
- 20 51. A method for transmitting a telephone stream generated by a source device on a network for reception in said network by a destination device, said method comprising:  
(a) expressing said telephone stream as a group of one or more symbols, wherein each symbol is represented by a scrambled multi-level electrical signal having at least three levels;  
25 (b) de-scrambling said scrambled multi-level electrical signal at said destination device to determine said group of one or more symbols;  
(c) grouping said one or more symbols into a symbol group to be decoded; and  
(d) decoding said symbol group to reconstruct said telephone stream at said destination device from at least one of said symbol groups.
- 30 52. The method of claim 51, wherein said multi-level electrical signal has five voltage levels.

53. The method of claim 51, wherein said symbol group is comprised of at least five symbols.
54. The method of claim 51, wherein one of said symbol groups represents at least two lanes of said telephone stream.
- 5 55. The method of claim 51, wherein at least two of said symbol groups represent at least four lanes of said telephone stream.
56. The method of claim 55, further comprising:  
appointing one of a plurality of devices connected to said network as a clock master;  
selecting one transmission mode from a plurality of transmission modes that is  
10 supported by a device of said plurality of devices and that is supported by a link connecting said device to said network; and  
configuring a loop in said network with substantially continuous back-to-back sequences of frames comprised of said at least four lanes of said telephone stream.
57. A method to determine a clock offset on a logical ring network having a plurality of  
15 devices, said method comprising:  
synchronizing a plurality of frame-counting clocks on said logical ring network by broadcasting a time mark command;  
specifying a frame count for each device of said plurality of devices that needs its  
frame count synchronized with other devices of said plurality of devices on said logical ring  
20 network;  
following said time mark command with the transmission of a marked frame that goes around said logical ring network;  
calculating a time difference value from said time mark command and said marked frame for at least one device of said plurality of devices; and  
25 transferring said time difference value into a frame-counting clock in said at least one device, wherein said time difference value is used to calculate a clock offset for said at least one device.
58. The method of claim 57, further comprising:  
appointing one of said devices connected to said logical ring network as a clock master  
30 during start-up of said logical ring network, or when one or more devices are added or removed from said logical ring network.

59. The method of claim 57, further comprising:  
selecting one transmission mode that is supported by a device of said plurality of devices and that is supported by a link connecting said device to said logical ring network.
60. A method for interfacing stream information between one or more network control protocols and a network physical layer, said method comprising:  
5 processing command and data in said stream information with a command stream processor;  
generating one or more network time and event signals with a network time and event generator;  
10 reading and writing command and data in said stream information communicated to said network physical layer on a physical layer interface;  
reading and writing serial data provided on a serial memory interface; and  
selectively resetting said command stream processor, said network time and event generator, said physical layer interface, and said serial memory interface.
- 15 61. The method of claim 60, further comprising:  
encoding and decoding video in said stream information communicated on a video bus;  
encoding and decoding audio in said stream information communicated on an audio connection;  
reading and writing general purpose digital input/output (GPIO) information  
20 communicated on a GPIO bus; and  
reading and writing host CPU information communicated on a host CPU bus.
62. The method of claim 60, further comprising:  
processing a plurality of asynchronous packets of data using a stream engine; and  
providing an asynchronous packet stream circuit.
- 25 63. A method for electing a device as a clock master from a plurality of devices on a logical ring network, said method comprising:  
selecting said first device of said plurality of devices as said clock master using an arbitration value;  
sending a first message on a plurality of ports from said first device, wherein if said  
30 first device receives an acknowledgment on all of said plurality of ports, said first device is elected said clock master, but if said first device receives a message from a second device containing a higher arbitration value than said arbitration value of said first device, said first

device sends a second message containing said higher arbitration value of said second device out on all remaining ports of said plurality of ports; and

sending a third message to said second device with said higher arbitration value as an acknowledgment message appointing said second device as said clock master, if an

- 5      acknowledgement is received on all of the other ports of said first device, or sending said third message to said second device with a higher arbitration value if a message with an even higher arbitration value than said arbitration value of said second device arrives on any of said plurality of ports of said first device.

64.      The method of claim 63, further comprising:

- 10      selecting said clock master based on an arbitration value, wherein one byte of said arbitration value prioritizes each device into one of a plurality of classes, and each device's identification number comprises a second byte of said arbitration value.

65.      A method for allocating a set of lanes in a frame containing a plurality of lanes, in a network connecting a plurality of devices, comprising:

- 15      transmitting a value from an originating device requesting said set of lanes in said frame, wherein said set of lanes are represented by a plurality of bits in said value;  
receiving said value at each device of said plurality of devices;  
removing a bit from said value for each corresponding lane of said plurality of lanes that said each device is using;  
20      receiving said value at said originating device; and  
setting a set of bits in a mask representing said set of lanes at said originating device, wherein said set of bits reserves said set of lanes for the use of said originating device.

66.      A method to transmit bi-directional synchronous data streams from a first device to a second device on a time-division multiplexed access (TDMA)-oriented network connecting

- 25      said first device and said second device, comprising:

allocating a first set of lanes in a frame containing a plurality of lanes to said first device when said first set of lanes can be allocated, wherein said frame is received by said second device;

- 30      allocating a second set of lanes in said frame to said second device when said second set of lanes can be allocated, wherein said frame is received by said first device;

transmitting a first group of synchronous data on said first set of lanes from said first device to said second device; and

transmitting a second group of synchronous data on said second set of lanes from said second device to said first device.

67. A method for broadcasting device identification during startup of each device in a network connecting a plurality of devices, comprising:

5 sending status information from said each device to the remainder of said network connecting said plurality of devices; and

sending device configuration information from each device to the remainder of said network connecting said plurality of devices.

68. A method for structuring the data architecture of a device read only memory (ROM) in a network connecting a plurality of devices, comprising:

10 assigning a first plurality of bytes in a device ROM of said device for one or more of the following purposes selected from the group consisting of: Protocol Version Number, Company ID, and Model ID;

15 assigning a second plurality of bytes in said device ROM of said device for one or more of the following purposes selected from the group consisting of: Protocol Hint Bit Mask, Capability Hint Bit Mask, and the number of streams implemented by the device; and

20 assigning a third plurality of bytes in said device ROM of said device for one or more of the following purposes selected from the group consisting of: start address of the device's Allocation Marker in a random access memory, start address of the device's Device Information in ROM, device status, logical loop number, and loop operating mode.

69. A network to communicate information by using symbols generated by a source device on said network for transmission in said network by said source device and reception in said network by a destination device, said network comprising:

25 an encoder to encode said information to produce an encoded symbol;  
a scrambler to express said encoded symbol as a scrambled multi-level electrical signal at said source device, wherein said multi-level electrical signal has at least three levels;

a de-scrambler to de-scramble said scrambled multi-level electrical signal at said destination device to determine said encoded symbol;

30 a buffer to group at least one of said encoded symbols into a symbol group; and  
a decoder to decode said symbol group into said information.

70. The network of claim 69, wherein said multi-level electrical signal has at least five levels.

71. The network of claim 69, wherein said symbol group is comprised of at least five symbols.
72. The network of claim 69, wherein said symbol group represents a plurality of binary bits.
- 5 73. The network of claim 69, wherein said symbol group represents nine binary bits.
74. The network of claim 69, wherein said symbol group is transmitted from said source device to said destination device on a connector selected from the following group, including a single pair of wires, a double pair of wires, a triple pair of wires, a coaxial cable, one optical fiber, and two optical fibers.
- 10 75. The network of claim 72, wherein said symbol group represents at least two lanes of information.
76. The network of claim 75, wherein a stream from said source device is carried by said at least two lanes of information.
77. The network of claim 75, wherein a data frame comprising said at least two lanes is  
15 propagated from said source device to said destination device.
78. The network of claim 77, wherein said data frame is preceded by a data frame marker detected in said scrambled multi-level electrical signal at said destination device.
79. The network of claim 69, wherein said symbol group contains a plurality of evenly spaced codes with a minimum distance of two between said codes to facilitate error detection  
20 and correction at said destination device.
80. The network of claim 79, further comprising an error detection and correction circuit, wherein the data in at least one stream packet is processed for error detection and error correction at said destination device.
81. A network to transmit a command stream generated by a source device for reception  
25 by a destination device in a network connecting a plurality of devices, comprising:  
an arbitration circuit to appoint and enable one of said plurality of devices as a clock master to provide a command stream token on said network, wherein each said source device on said network is required to receive said command stream token before transmission of said

command stream, and hold said command stream token until transmission of said command stream is complete;

an encoder to encode said command stream as a group of one or more symbols, wherein each symbol is represented by a multi-level electrical signal having at least three  
5 levels;

a receiving circuit to receive at said destination device one or more said multi-level electrical signals representing said group of one or more symbols; and

a decoder to decode said group of one or more symbols to reconstruct said command stream at said destination device.

10 82. The network of claim 81, further comprising an arbitration circuit to appoint one of said plurality of devices as said clock master at a start-up of said network, or when one or more devices are added or removed from said network.

83. The network of claim 81, further comprising:

means for performance testing said plurality of devices at a start-up of said network  
15 using a transmission mode from one or more transmission modes; and

means for selecting one transmission mode from one or more transmission modes that is supported by said plurality of devices.

84. The network of claim 81, wherein said multi-level electrical signal has at least five voltage levels.

20 85. The network of claim 81, wherein said group of one or more symbols is comprised of at least five symbols.

86. The network of claim 81, wherein said group of one or more symbols represents a plurality of lanes of information.

87. The network of claim 81, wherein at least two of said groups of one or more symbols  
25 represent said command stream.

88. The network of claim 81, wherein said command stream is comprised of a variable number of said groups of one or more symbols.

89. The network of claim 81, wherein said decoder comprises decode logic and decodes said command stream into at least one control code.



90. A network for transmitting an audio stream generated by a source device on a network for reception in said network by a destination device, comprising:
- a scrambler to express said audio stream as a group of two or more symbols, wherein each symbol is represented by a scrambled multi-level electrical signal having at least three levels;
  - a de-scrambler to de-scramble and error correct said scrambled multi-level electrical signal at said destination device to determine said group of two or more symbols;
  - a buffer to group said one or more symbols into a symbol group to be decoded; and
  - a decoder to decode said symbol group to reconstruct said audio stream at said destination device from at least two of said symbol groups.
91. The network of claim 90, wherein said multi-level electrical signal has five voltage levels.
92. The network of claim 90, wherein said symbol group is comprised of at least five symbols.
93. The network of claim 90, wherein said symbol group represents a plurality of lanes of audio information.
94. The network of claim 93, further comprising:
- an arbitration circuit to appoint one of a plurality of devices connected to said communication network as a clock master; and
  - a configuration circuit to select a transmission mode from one or more transmission modes that is supported by a device of said plurality of devices and that is supported by a link connecting said device to said communication network.
95. The network of claim 90, wherein said audio stream is transmitted from said source device to said destination device at a substantially constant network frame rate.
96. The network of claim 95, wherein said audio stream is frame synchronous, wherein one audio packet is contained in a network frame and the respective nibbles of each audio packet are mapped onto the same lane in every said network frame.
97. The network of claim 95, wherein said audio stream is clock synchronous.

98. The network of claim 95, wherein said audio stream is running asynchronously at a sampling rate extracted from an external clock independent of said network frame rate.

99. A network for transmitting an asynchronous packet stream generated by a source device on a network for reception in said network by a destination device, comprising:

5 a scrambler to express said asynchronous packet stream as a group of one or more symbols, wherein each symbol is represented by a scrambled multi-level electrical signal having at least three levels;

a de-scrambler to de-scramble said scrambled multi-level electrical signal at said destination device to determine said group of one or more symbols;

10 a buffer to group said one or more symbols into a symbol group to be decoded; and

a decoder to decode said symbol group to reconstruct said asynchronous packet stream at said destination device from at least two of said symbol groups.

100. The network of claim 99, wherein said multi-level electrical signal has five voltage levels.

15 101. The network of claim 99, wherein said symbol group is comprised of at least five symbols.

102. The network of claim 99, wherein said symbol group represents a plurality of lanes of information.

20 103. The network of claim 99, wherein said asynchronous packet stream is comprised of a variable number of symbol groups.

104. The network of claim 95, wherein said asynchronous packet stream contains a plurality of evenly spaced codes with a minimum distance of two between said codes to facilitate error detection and correction at said destination device.

25 105. The network of claim 102, wherein the data in at least one asynchronous packet stream packet is processed for error detection and error correction at said destination device.

106. The network of claim 102, further comprising:

an arbitration circuit to appoint one of a plurality of devices connected to said communication network as a clock master.

107. The network of claim 102, further comprising:

means for performance testing said plurality of devices at a start-up of said network using a transmission mode from one or more transmission modes; and

a configuration circuit to select one transmission mode from said one or more transmission modes that is supported by said plurality of devices; and

5 a loop in said network with substantially continuous back-to-back sequences of frames comprised of said plurality of lanes.

108. A network to transmit a telephone stream generated by a source device on a network for reception in the network by a destination device, comprising:

a scrambler in said source device to express said telephone stream as a group of one or  
10 more symbols, wherein each symbol is represented by a scrambled multi-level electrical signal having at least three levels;

a de-scrambler to de-scramble said scrambled multi-level electrical signal at said destination device to determine said group of one or more symbols;

a buffer to group said one or more symbols into a symbol group to be decoded; and

15 a decoder to decode said symbol group to reconstruct said telephone stream at said destination device from at least two of said symbol groups.

109. The network of claim 108, wherein said multi-level electrical signal has five voltage levels.

110. The network of claim 108, wherein said symbol group is comprised of at least five  
20 symbols.

111. The network of claim 108, wherein said symbol group represents a plurality of lanes of information.

112. The network of claim 108, wherein said at least two symbol groups represent at least four lanes of said telephone stream.

25 113. The network of claim 112, further comprising:

an arbitration circuit to appoint one of a plurality of devices connected to said network as a clock master;

a configuration circuit to select the fastest transmission mode that is supported by a device of said plurality of devices and that is supported by a link connecting said device to  
30 said network; and

a loop in said network with substantially continuous back-to-back sequences of frames comprised of said at least four lanes of said telephone stream.

114. A mechanism to determine a clock offset on a logical ring network having a plurality of devices, comprising:

- 5 means for synchronizing a plurality of frame-counting clocks on said logical ring network by broadcasting a time mark command;
- means for specifying a frame count for each device of said plurality of devices that needs its frame count synchronized with other devices of said plurality of devices on said logical ring network;
- 10 means for following said time mark command with the transmission of a marked frame that goes around said logical ring network;
- means for calculating a time difference value from said time mark command and said marked frame for at least one device of said plurality of devices; and
- means for transferring said time difference value into a clock in said at least one
- 15 device, wherein said time difference value is used to calculate a clock offset for said at least one device.

115. A media access controller circuit for interfacing stream information between one or more network control protocols and a network physical layer, comprising:

- a command stream processor to process command and data in said stream information;
- 20 a physical layer interface to read and write command and data in said stream information to said network physical layer;
- a serial memory interface to read and write serial data;
- a network time and event generator to generate network time and event information;
- and
- 25 a reset circuit to selectively reset part or all of said media access controller.

116. The media access controller of claim 115, further comprising:

- a video stream encoder/decoder to encode and decode video data received or sent on a video bus;
- an audio stream encoder/decoder to encode and decode audio data received or sent on
- 30 an audio bus;
- a general purpose digital input/output (GPIO) bus to read and write GPIO information;
- and

a host CPU bus to read and write host CPU information.

117. The media access controller of claim 115, further comprising:

a stream engine to operate on asynchronous packets of data;

an asynchronous packet stream circuit; and

5 a synchronous direct memory access circuit to transfer data between said media access controller and a memory.

118. A clock arbitration mechanism to elect a device as a clock master from a plurality of devices on a logical ring network, comprising:

means for selecting said first device of said plurality of devices as said clock  
10 master using an arbitration value;

a plurality of ports for sending a first message from said first device, wherein if said first device receives an acknowledgment on all of said plurality of ports, said first device is elected said clock master, but if said first device receives a message from a second device containing a higher arbitration value than said arbitration value of said first device, said first  
15 device sends a second message containing said higher arbitration value of said second device out on all remaining ports of said plurality of ports; and

means for sending a third message to said second device with said higher arbitration value as an acknowledgment message appointing said second device as said clock master, if an acknowledgment is received on all of the other ports of said first device, or sending said third  
20 message to said second device with a higher arbitration value if a message with an even higher arbitration value than said arbitration value of said second device arrives on any of said plurality of ports of said first device.

119. The clock arbitration mechanism of claim 118, further comprising:

means for selecting said clock master based on an arbitration value, wherein one  
25 byte of said arbitration value prioritizes each device into one of a plurality of classes, and each device's identification number comprises a second byte of said arbitration value.

120. A lane allocation mechanism to allocate a set of lanes in a frame containing a plurality of lanes, in a network connecting a plurality of devices, comprising:

means for transmitting a value from an originating device requesting said set of lanes  
30 in said frame, wherein said set of lanes are represented by a plurality of bits in said value;  
means for receiving said value at each device of said plurality of devices;

means for removing a bit from said value for each corresponding lane of said plurality of lanes that said each device is using;

means for receiving said value at said originating device; and

means for setting a set of bits in a mask representing said set of lanes at said  
5 originating device, wherein said set of bits reserves said set of lanes for the use of said originating device.

121. A TDMA-oriented network connecting a plurality of devices to transmit bi-directional synchronous data streams from a first device to a second device, comprising:

means for receiving a request from said first device to allocate a first plurality of lanes;

10 means for allocating said first plurality of lanes to said first device if said first plurality of lanes can be allocated;

means for transmitting synchronous data on said first plurality of lanes from said first device to said second device;

means for receiving a request from said second device to allocate a second plurality of  
15 lanes;

means for allocating said second plurality of lanes to said second device if said second plurality of lanes can be allocated; and

means for transmitting synchronous data on said second plurality of lanes from said second device to said first device.

20 122. A structure for device identification during startup of a device in a network connecting a plurality of devices, comprising:

status information concerning said device, wherein said status information is sent to the remainder of said network connecting said plurality of devices; and

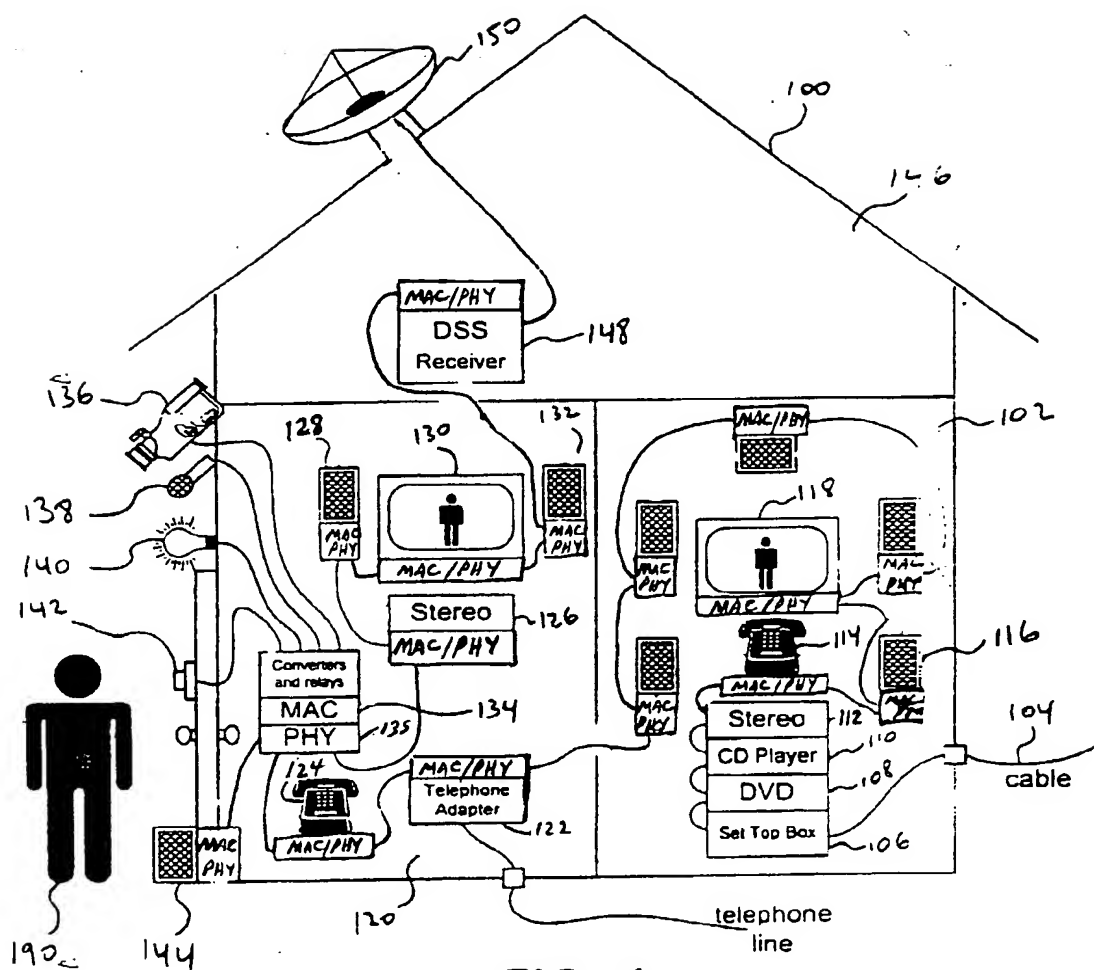
device configuration information concerning said device, wherein said device  
25 configuration information is sent to the remainder of said network connecting said plurality of devices.

123. A data architecture of a device read only memory (ROM) in a network connecting a plurality of devices, comprising:

a first plurality of bytes in a device ROM of said device reserved for one or more of the  
30 following purposes selected from the group consisting of: Protocol Version Number, Company ID, and Model ID;

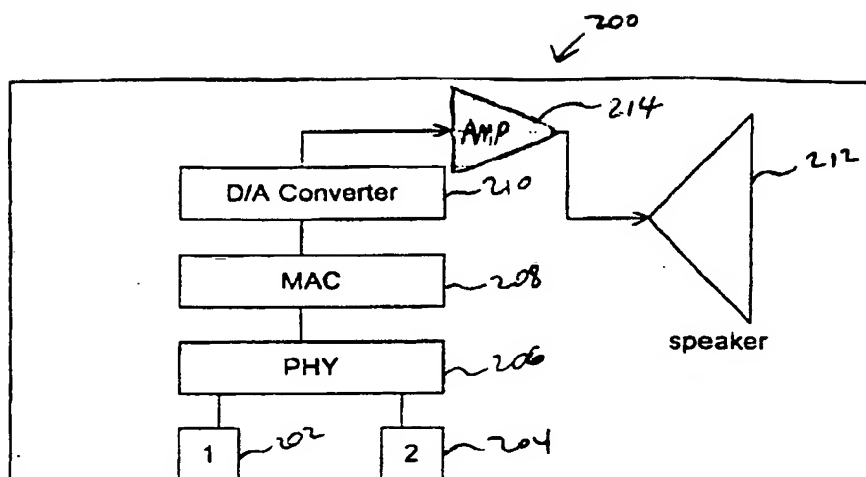
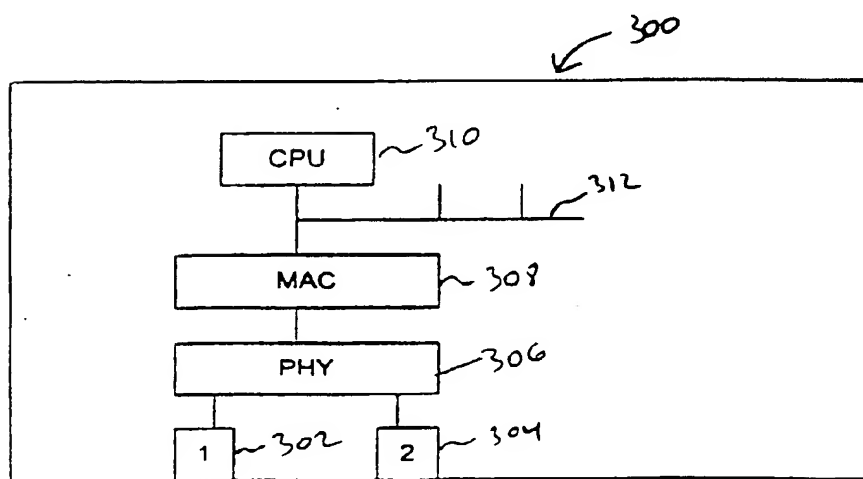
a second plurality of bytes in said device ROM of said device reserved for one or more of the following purposes selected from the group consisting of: Protocol Hint Bit Mask, Capability Hint Bit Mask, and the number of streams implemented by the device; and

- a third plurality of bytes in said device ROM of said device reserved for one or more of
- 5 the following purposes selected from the group consisting of: start address of the device's Allocation Marker in a random access memory, start address of the device's Device Information in ROM, device status, logical loop number, and loop operating mode.



**FIG. 1**



**FIG. 2****FIG. 3**

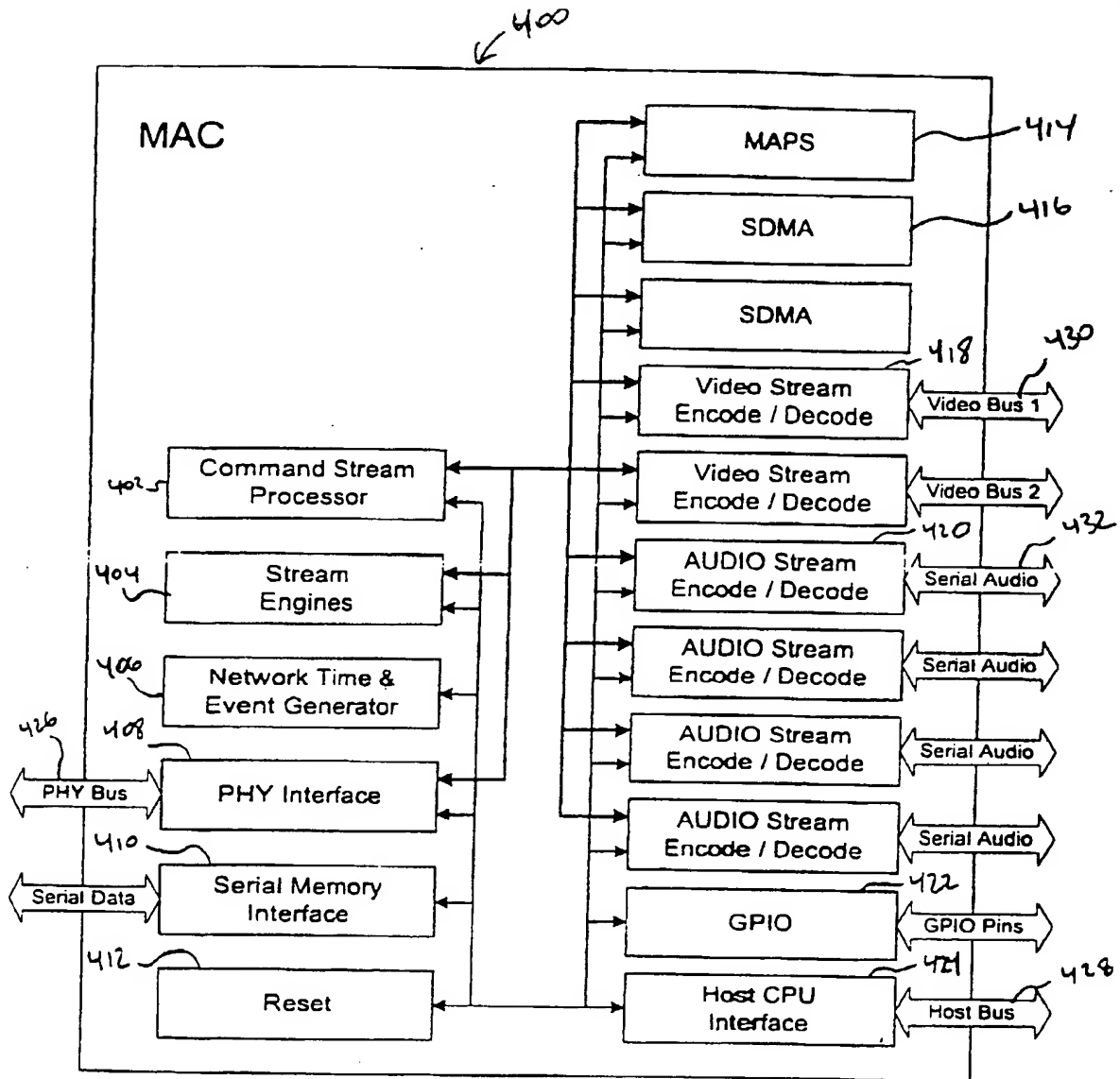
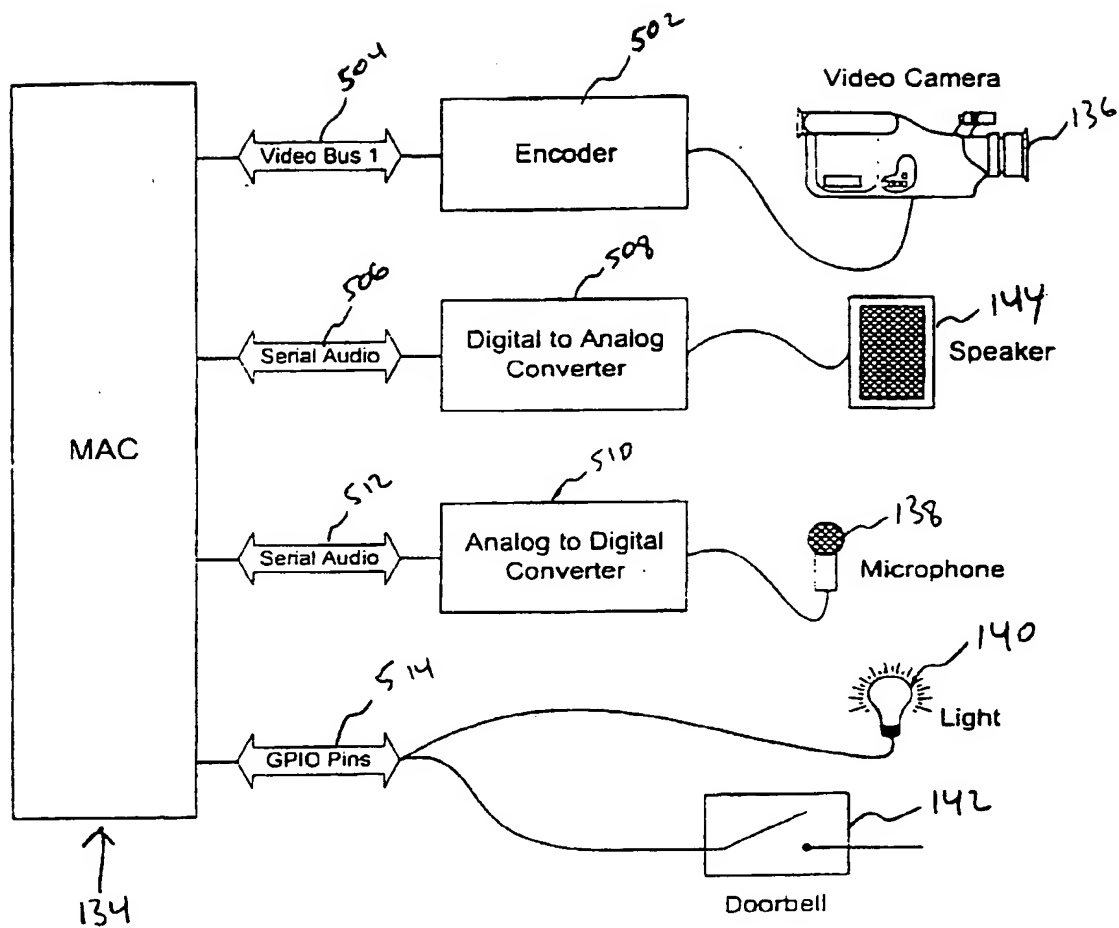


FIG. 4

**FIG. 5**

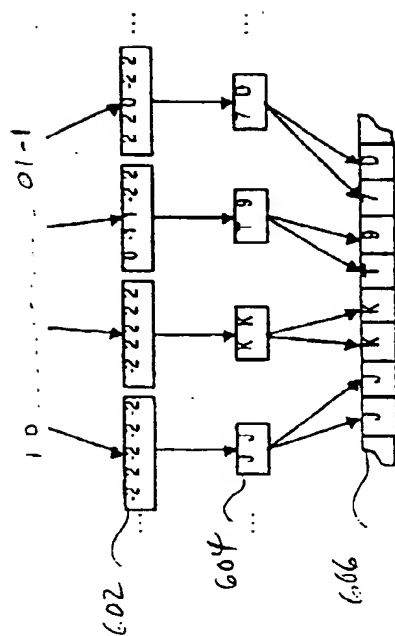


FIG. 6

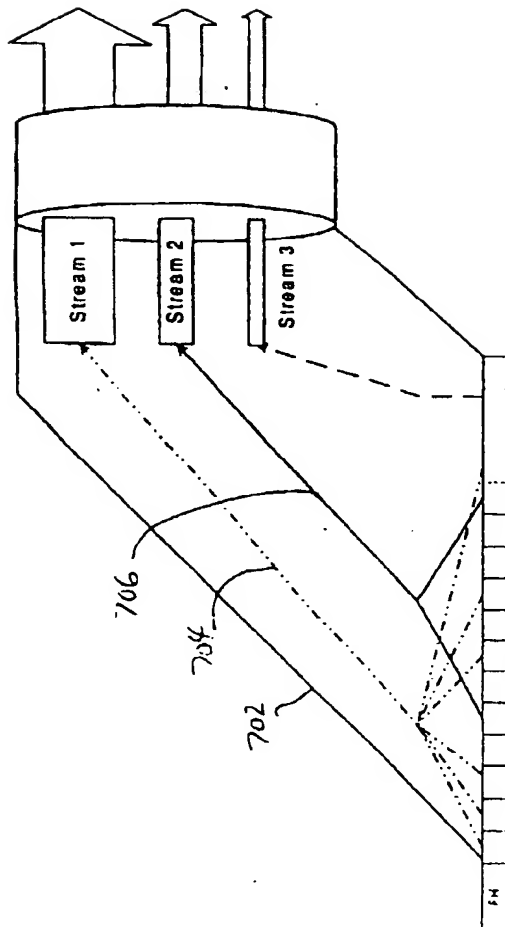


FIG. 7

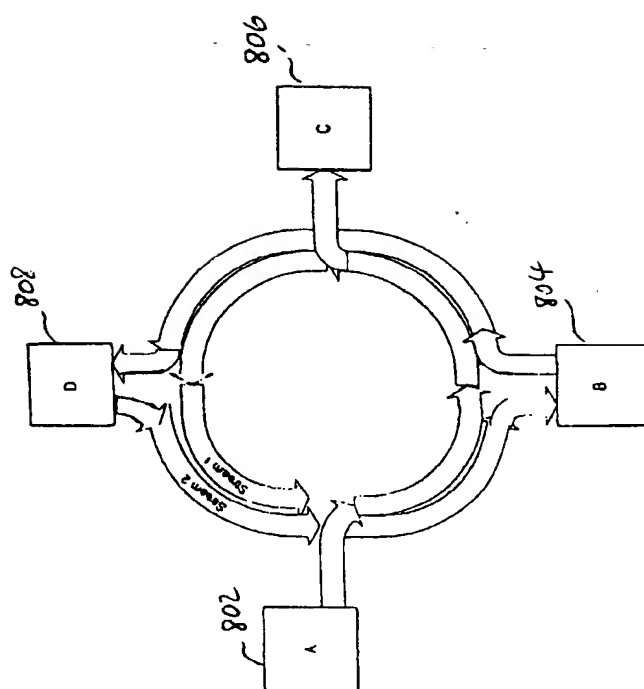


FIG. 8

900

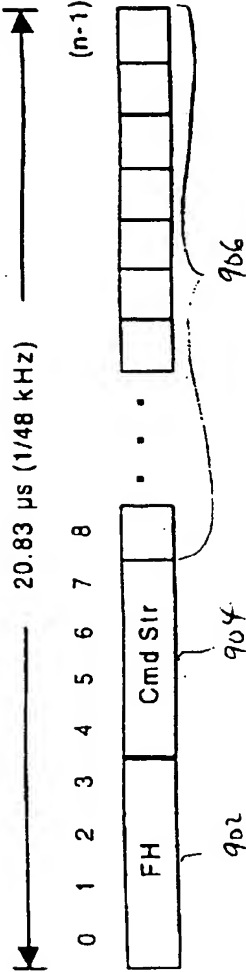


FIG. 9

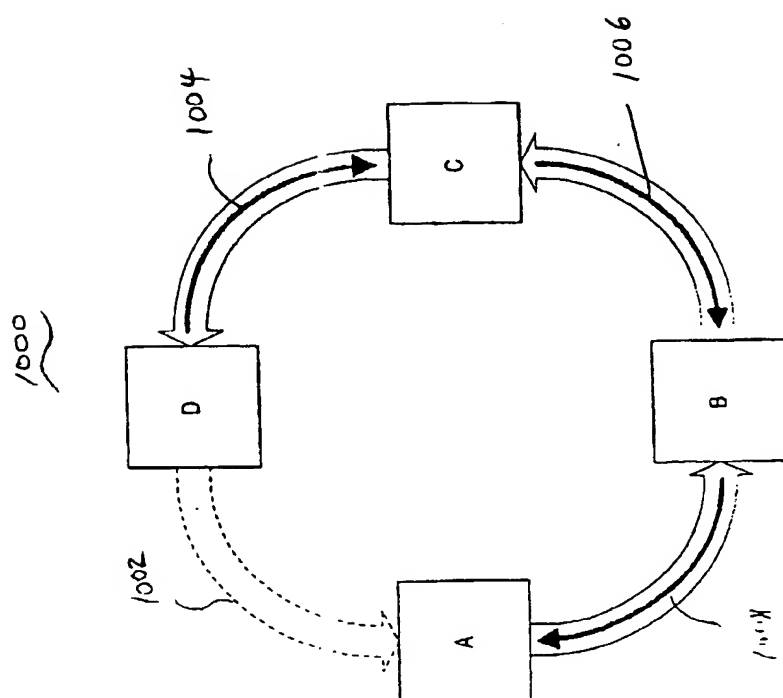


FIG. 10



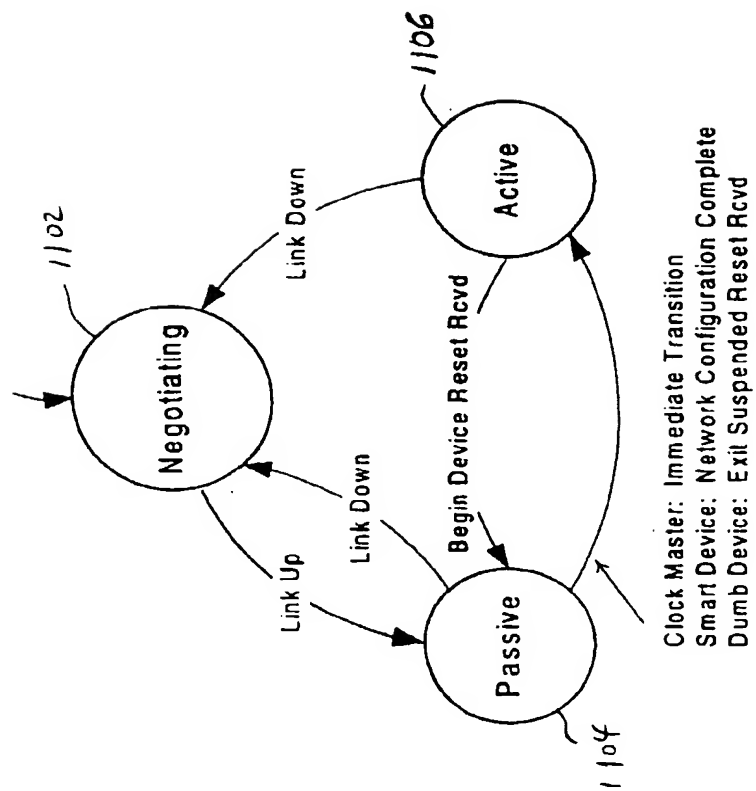


FIG. 11

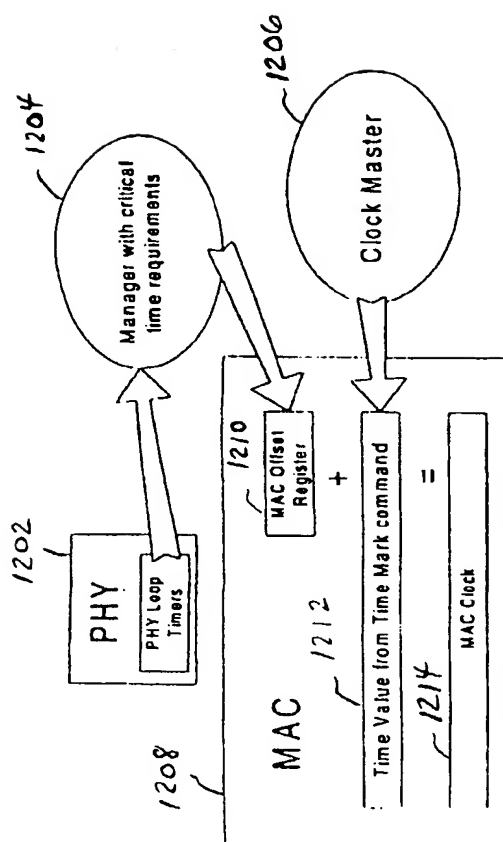


FIG. 12

12/20

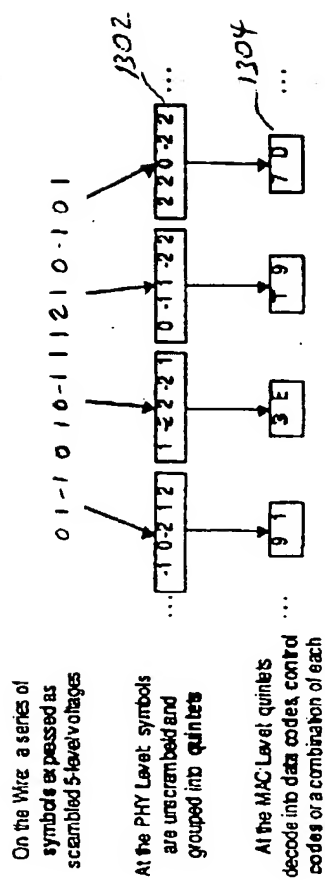


FIG. 13

Index	y[1]	y[2]	y[3]	y[4]	y[5]	Hex MS Nibble	Hex LS Nibble
0	1	-1	-2	0	2	00	00
1	1	-1	-2	1	1	01	01
2	1	-1	-2	2	0	02	02
3	1	-1	-1	-1	2	03	03
4	1	-1	-1	0	1	04	04
5	1	-1	-1	1	0	05	05
6	1	-1	-1	2	-1	06	06
7	1	-1	0	-2	2	07	07
8	1	-1	0	-1	1	08	08
9	1	-1	0	0	0	09	09
10	1	-1	0	1	-1	0A	0A
11	1	-1	0	2	-2	0B	0B
12	1	-1	1	-2	1	0C	0C
13	1	-1	1	-1	0	0D	0D
14	1	-1	1	0	-1	0E	0E
15	1	-1	1	1	-2	0F	0F
16	1	0	2	-1	-2	10	10
17	1	0	2	-2	-1	11	11
18	1	0	1	0	-2	12	12
19	1	0	1	-1	-1	13	13
20	1	0	1	-2	0	14	14
21	1	0	0	1	-2	15	15
22	1	0	0	0	-1	16	16
23	1	0	0	-1	0	17	17
24	1	0	0	-2	1	18	18
25	1	0	-1	2	-2	19	19
26	1	0	-1	1	-1	1A	1A
27	1	0	-1	0	0	1B	1B
28	1	0	-1	-1	1	1C	1C
29	1	0	-1	-2	2	1D	1D
30	1	0	-2	2	-1	1E	1E
31	1	0	-2	1	0	1F	1F
32	1	1	2	-2	-2	20	20
33	1	1	1	-1	-2	21	21
34	1	1	1	-2	-1	22	22
35	1	1	0	0	-2	23	23
36	1	1	0	-1	-1	24	24
37	1	1	0	-2	0	25	25
38	1	1	-1	1	-2	26	26
39	1	1	-1	0	-1	27	27
40	1	1	-1	-1	0	28	28
41	1	1	-1	-2	1	29	29
42	1	1	-2	2	-2	2A	2A
43	1	1	-2	1	-1	2B	2B
44	1	1	-2	0	0	2C	2C
45	1	1	-2	-1	1	2D	2D
46	1	1	-2	-2	2	2E	2E
47	1	-1	2	-2	0	2F	2F
48	1	-2	-2	1	2	30	30
49	1	-2	-2	2	1	31	31
50	1	-2	-1	0	2	32	32
51	1	-2	-1	1	1	33	33
52	1	-2	-1	2	0	34	34
53	1	-2	0	-1	2	35	35
54	1	-2	0	0	1	36	36
55	1	-2	0	1	0	37	37
56	1	-2	0	2	-1	38	38
57	1	-2	1	-2	2	39	39
58	1	-2	1	-1	1	3A	3A
59	1	-2	1	0	0	3B	3B
60	1	-2	1	1	-1	3C	3C
61	1	-2	1	2	-2	3D	3D
62	1	-2	2	-2	1	3E	3E
63	1	-2	2	-1	0	3F	3F
64	2	-2	-2	0	2	40	40

Index	y[1]	y[2]	y[3]	y[4]	y[5]	Hex MS Nibble	Hex LS Nibble
65	2	-2	-2	1	1	41	41
66	2	-2	-2	2	0	42	42
67	2	-2	-1	-1	2	43	43
68	2	-2	-1	0	1	44	44
69	2	-2	-1	1	0	45	45
70	2	-2	-1	2	-1	46	46
71	2	-2	0	-2	2	47	47
72	2	-2	0	-1	1	48	48
73	2	-2	0	0	0	49	49
74	2	-2	0	1	-1	4A	4A
75	2	-2	0	2	-2	4B	4B
76	2	-2	1	-2	1	4C	4C
77	2	-2	1	-1	0	4D	4D
78	2	-2	1	0	-1	4E	4E
79	2	-2	1	1	-2	4F	4F
80	2	-1	2	-1	-2	50	50
81	2	-1	2	-2	-1	51	51
82	2	-1	1	0	-2	52	52
83	2	-1	1	-1	-1	53	53
84	2	-1	1	-2	0	54	54
85	2	-1	0	1	-2	55	55
86	2	-1	0	0	-1	56	56
87	2	-1	0	-1	0	57	57
88	2	-1	0	-2	1	58	58
89	2	-1	-1	2	-2	59	59
90	2	-1	-1	1	-1	5A	5A
91	2	-1	-1	0	0	5B	5B
92	2	-1	-1	-1	1	5C	5C
93	2	-1	-1	-2	2	5D	5D
94	2	-1	-2	2	-1	5E	5E
95	2	-1	-2	1	0	5F	5F
96	2	0	2	-2	-2	60	60
97	2	0	1	-1	-2	61	61
98	2	0	1	-2	-1	62	62
99	2	0	0	0	-2	63	63
100	2	0	0	-1	-1	64	64
101	2	0	0	-2	0	65	65
102	2	0	-1	1	-2	66	66
103	2	0	-1	0	-1	67	67
104	2	0	-1	-1	0	68	68
105	2	0	-1	-2	1	69	69
106	2	0	-2	2	-2	6A	6A
107	2	0	-2	1	-1	6B	6B
108	2	0	-2	0	0	6C	6C
109	2	0	-2	-1	1	6D	6D
110	2	0	-2	-2	2	6E	6E
111	2	-2	2	-2	0	6F	6F
112	2	1	1	-2	-2	70	70
113	2	1	0	-1	-2	71	71
114	2	1	0	-2	-1	72	72
115	2	1	-1	0	-2	73	73
116	2	1	-1	-1	-1	74	74
117	2	1	-1	-2	0	75	75
118	2	1	-2	1	-2	76	76
119	2	1	-2	0	-1	77	77
120	2	1	-2	-1	0	78	78
121	2	1	-2	-2	1	79	79
122	2	2	0	-2	-2	7A	7A
123	2	2	-1	-1	-2	7B	7B
124	2	2	-1	-2	-1	7C	7C
125	2	2	-2	0	-2	7D	7D
126	2	2	-2	-1	-1	7E	7E
127	2	2	-2	-2	0	7F	7F
128	-1	1	-2	0	2	80	80
129	-1	1	-2	1	1	81	81

FIG. 14(a)

14/20

Index	y[1]	y[2]	y[3]	y[4]	y[5]	Hex MS Nibble	Hex LS Nibble
130	-1	1	-2	2	0	82	C1
131	-1	1	-1	-1	2	83	C2
132	-1	1	-1	0	1	84	C3
133	-1	1	-1	1	0	85	C4
134	-1	1	-1	2	-1	86	C5
135	-1	1	0	-2	2	87	C6
136	-1	1	0	-1	1	88	C7
137	-1	1	0	0	0	89	C8
138	-1	1	0	1	-1	8A	C9
139	-1	1	0	2	-2	8B	CA
140	-1	1	1	-2	1	8C	CB
141	-1	1	1	-1	0	8D	CC
142	-1	1	1	0	-1	8E	CD
143	-1	1	1	1	-2	8F	CE
144	-1	0	-2	1	2	90	CF
145	-1	0	-2	2	1	91	C0
146	-1	0	-1	0	2	92	D1
147	-1	0	-1	1	1	93	D2
148	-1	0	-1	2	0	94	D3
149	-1	0	0	-1	2	95	D4
150	-1	0	0	0	1	96	D5
151	-1	0	0	1	0	97	D6
152	-1	0	0	2	-1	98	D7
153	-1	0	1	-2	2	99	D8
154	-1	0	1	-1	1	9A	D9
155	-1	0	1	0	0	9B	DA
156	-1	0	1	1	-1	9C	DB
157	-1	0	1	2	-2	9D	DC
158	-1	0	2	-2	1	9E	DD
159	-1	0	2	-1	0	9F	DE
160	-1	-1	-2	2	2	AA	DF
161	-1	-1	-1	1	2	A1	E0
162	-1	-1	-1	2	1	A2	E1
163	-1	-1	0	0	2	A3	E2
164	-1	-1	0	1	1	A4	E3
165	-1	-1	0	2	0	A5	E4
166	-1	-1	1	-1	2	A6	E5
167	-1	-1	1	0	1	A7	E6
168	-1	-1	1	1	0	A8	E7
169	-1	-1	1	2	-1	A9	E8
170	-1	-1	2	-2	2	AA	E9
171	-1	-1	2	-1	1	AB	EA
172	-1	-1	2	0	0	AC	EB
173	-1	-1	2	1	-1	AD	EC
174	-1	-1	2	2	-2	AE	ED
175	-1	1	2	-2	0	AF	EE
176	-1	2	2	-1	-2	B0	EF
177	-1	2	2	-2	-1	B1	F0
178	-1	2	1	0	-2	B2	F1
179	-1	2	1	-1	-1	B3	F2
180	-1	2	1	-2	0	B4	F3
181	-1	2	0	1	-2	B5	F4
182	-1	2	0	0	-1	B6	F5
183	-1	2	0	-1	0	B7	F6
184	-1	2	0	-2	1	B8	F7
185	-1	2	-1	2	-2	B9	F8
186	-1	2	-1	1	-1	BA	F9
187	-1	2	-1	0	0	BB	FA
188	-1	2	-1	-1	1	BC	FB
189	-1	2	-1	-2	2	BD	FC
190	-1	2	-2	2	-1	BE	FD
191	-1	2	-2	1	0	BF	FE
192	-2	2	-2	0	2	CC	FF

Index	y[1]	y[2]	y[3]	y[4]	y[5]	Hex MS Nibble	Hex LS Nibble
193	-2	2	-2	1	1	C1	C1
194	-2	2	-2	2	0	C2	C2
195	-2	2	-1	-1	2	C3	C3
196	-2	2	-1	0	1	C4	C4
197	-2	2	-1	1	0	C5	C5
198	-2	2	-1	2	-1	C6	C6
199	-2	2	0	-2	2	C7	C7
200	-2	2	0	-1	1	C8	C8
201	-2	2	0	0	0	C9	C9
202	-2	2	0	1	-1	CA	CA
203	-2	2	0	2	-2	CB	CB
204	-2	2	1	-2	1	CC	CC
205	-2	2	1	-1	0	CD	CD
206	-2	2	1	0	-1	CE	CE
207	-2	2	1	1	-2	CF	CF
208	-2	1	-2	1	2	C0	C0
209	-2	1	-2	2	1	D1	D1
210	-2	1	-1	0	2	D2	D2
211	-2	1	-1	1	1	D3	D3
212	-2	1	-1	2	0	D4	D4
213	-2	1	0	-1	2	D5	D5
214	-2	1	0	0	1	D6	D6
215	-2	1	0	1	0	D7	D7
216	-2	1	0	2	-1	D8	D8
217	-2	1	1	-2	2	D9	D9
218	-2	1	1	-1	1	DA	DA
219	-2	1	1	0	0	DB	DB
220	-2	1	1	1	-1	DC	DC
221	-2	1	1	2	-2	DD	DD
222	-2	1	2	-2	1	DE	DE
223	-2	1	2	-1	0	DF	DF
224	-2	0	-2	2	2	E0	E0
225	-2	0	-1	1	2	E1	E1
226	-2	0	-1	2	1	E2	E2
227	-2	0	0	0	2	E3	E3
228	-2	0	0	1	1	E4	E4
229	-2	0	0	2	0	E5	E5
230	-2	0	1	-1	2	E6	E6
231	-2	0	1	0	1	E7	E7
232	-2	0	1	1	0	E8	E8
233	-2	0	1	2	-1	E9	E9
234	-2	0	2	-2	2	EA	EA
235	-2	0	2	-1	1	EB	EB
236	-2	0	2	0	0	EC	EC
237	-2	0	2	1	-1	ED	ED
238	-2	0	2	2	-2	EE	EE
239	-2	2	2	-2	0	EF	EF
240	-2	-1	-1	2	2	F0	F0
241	-2	-1	0	1	2	F1	F1
242	-2	-1	0	2	1	F2	F2
243	-2	-1	1	0	2	F3	F3
244	-2	-1	1	1	1	F4	F4
245	-2	-1	1	2	0	F5	F5
246	-2	-1	2	-1	2	F6	F6
247	-2	-1	2	0	1	F7	F7
248	-2	-1	2	1	0	F8	F8
249	-2	-1	2	2	-1	F9	F9
250	-2	-2	0	2	2	FA	FA
251	-2	-2	1	1	2	FB	FB
252	-2	-2	1	2	1	FC	FC
253	-2	-2	2	0	2	FD	FD
254	-2	-2	2	1	1	FE	FE
255	-2	-2	2	2	0	FF	FF

FIG. 14(b)

15/20

Code index	y[1]	y[2]	y[3]	y[4]	y[5]	MS Half Quintet	LS Half Quintet
256	0	1	2	-1	-2	'I'	0
257	0	1	2	-2	-1	'I'	1
258	0	1	1	0	-2	'I'	2
259	0	1	1	-1	-1	'I'	3
260	0	1	1	-2	0	'I'	4
261	0	1	0	1	-2	'I'	5
262	0	1	0	0	-1	'I'	6
263	0	1	0	-1	0	'I'	7
264	0	1	0	-2	1	'I'	8
265	0	1	-1	2	-2	'I'	9
266	0	1	-1	1	-1	'I'	A
267	0	1	-1	0	0	'I'	B
268	0	1	-1	-1	1	'I'	C
269	0	1	-1	-2	2	'I'	D
270	0	1	-2	2	-1	'I'	E
271	0	1	-2	1	0	'I'	F
272	0	2	2	-2	-2	0	'I'
273	0	2	1	-1	-2	1	'I'
274	0	2	1	-2	-1	2	'I'
275	0	2	0	0	-2	3	'I'
276	0	2	0	-1	-1	4	'I'
277	0	2	0	-2	0	5	'I'
278	0	2	-1	1	-2	6	'I'
279	0	2	-1	0	-1	7	'I'
280	0	2	-1	-1	0	8	'I'
281	0	2	-1	-2	1	9	'I'
282	0	2	-2	2	-2	A	'I'
283	0	2	-2	1	-1	B	'I'
284	0	2	-2	0	0	C	'I'
285	0	2	-2	-1	1	D	'I'
286	0	2	-2	-2	2	E	'I'
287	0	0	2	-2	0	F	'I'
288	0	-1	-2	1	2	'T'	0
289	0	-1	-2	2	1	'T'	1
290	0	-1	-1	0	2	'T'	2
291	0	-1	-1	1	1	'T'	3
292	0	-1	-1	2	0	'T'	4
293	0	-1	0	-1	2	'T'	5
294	0	-1	0	0	1	'T'	6
295	0	-1	0	1	0	'T'	7
296	0	-1	0	2	-1	'T'	8

Code index	y[1]	y[2]	y[3]	y[4]	y[5]	MS Half Quintet	LS Half Quintet
297	0	-1	1	-2	2	'T'	9
298	0	-1	1	-1	1	'T'	A
299	0	-1	1	0	0	'T'	B
300	0	-1	1	1	-1	'T'	C
301	0	-1	1	2	-2	'T'	D
302	0	-1	2	-2	1	'T'	E
303	0	-1	2	-1	0	'T'	F
304	0	-2	-2	2	2	0	'T'
305	0	-2	-1	1	2	1	'T'
306	0	-2	-1	2	1	2	'T'
307	0	-2	0	0	2	3	'T'
308	0	-2	0	1	1	4	'T'
309	0	-2	0	2	0	5	'T'
310	0	-2	1	-1	2	6	'T'
311	0	-2	1	0	1	7	'T'
312	0	-2	1	1	0	8	'T'
313	0	-2	1	2	-1	9	'T'
314	0	-2	2	-2	2	A	'T'
315	0	-2	2	-1	1	B	'T'
316	0	-2	2	0	0	C	'T'
317	0	-2	2	1	-1	D	'T'
318	0	-2	2	2	-2	E	'T'
319	0	0	1	-1	0	F	'T'
320	0	0	-2	0	2	'I'	'I'
321	0	0	-2	1	1	'I'	'T'
322	0	0	-2	2	0	'T'	'I'
323	0	0	-1	-1	2	'T'	'T'
324	0	0	-1	0	1	'R'	'R'
325	0	0	-1	1	0	Reserved	
326	-2	2	-2	-2	-2	'J'	'J'
327	-2	2	2	2	2	inverted	inverted
328	-2	2	2	2	2	'J'	'J'
329	2	-2	-2	-2	-2	inverted	inverted
330	-1	1	-2	-2	-2	'W'	'W'
331	1	-1	2	2	2	inverted	inverted
						'W'	'W'

FIG. 15

16/20

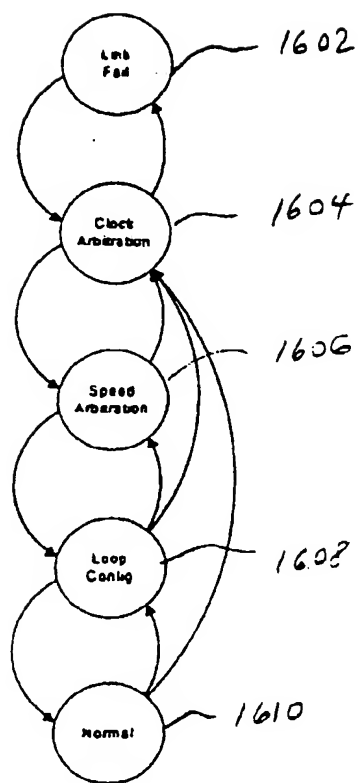


FIG. 16

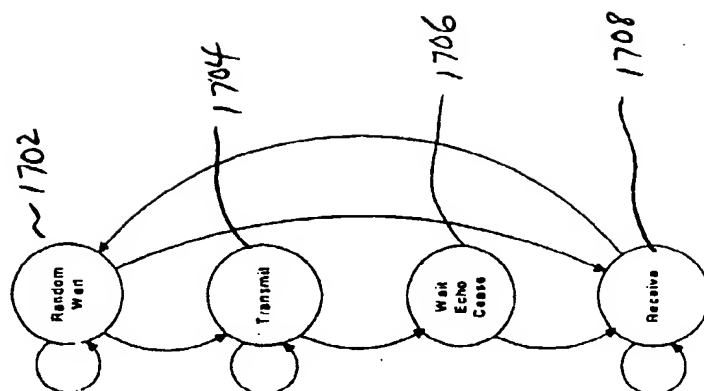


FIG. 17



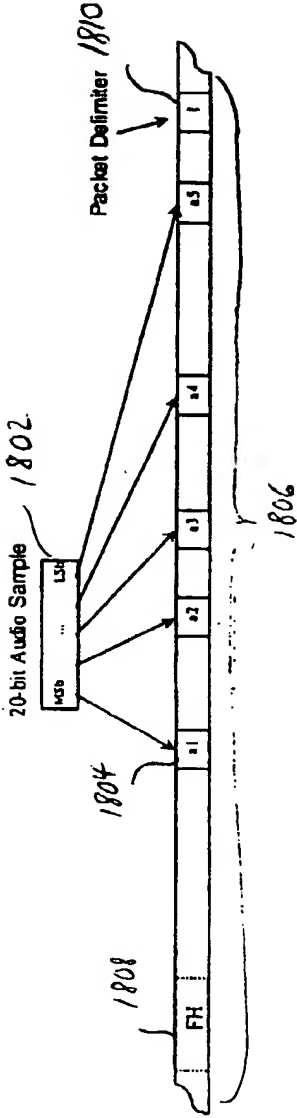


FIG. 18

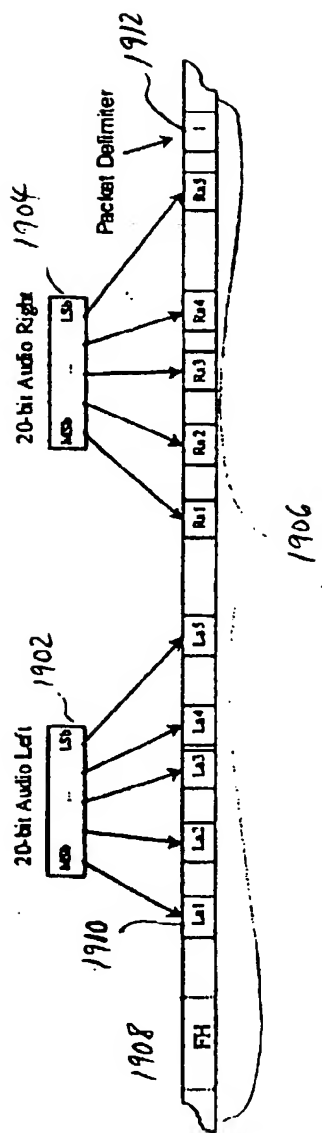


FIG. 19

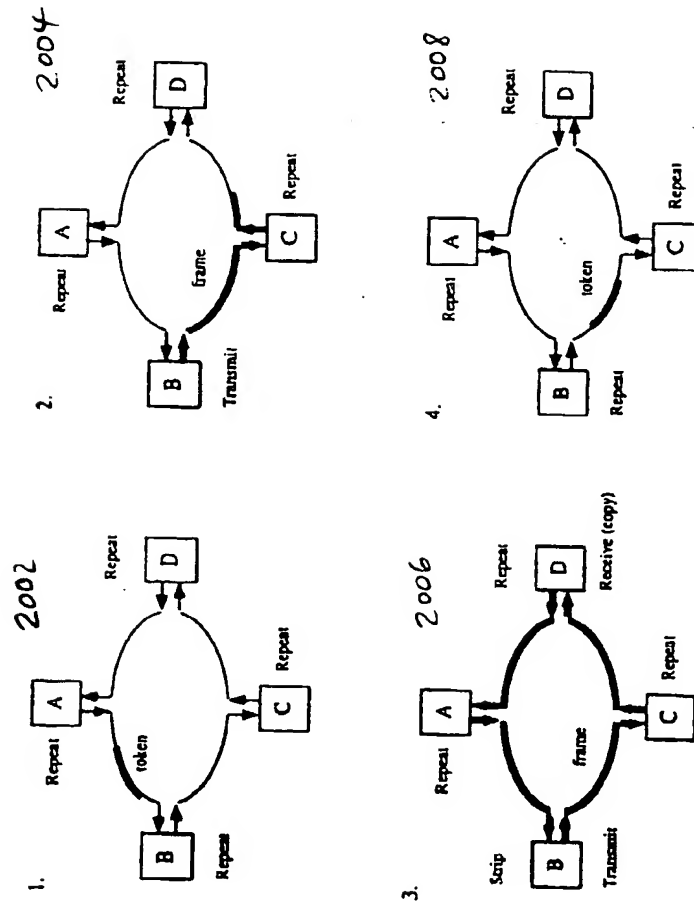


FIG. 20

# INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 00/19979

## A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 H04L25/49

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, INSPEC, COMPENDEX, IBM-TDB

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 97 34397 A (COLES ALISTAIR NEIL ;MOWBRAY MIRANDA JANE FELICITY (GB); CROUCH SI) 18 September 1997 (1997-09-18) abstract page 3, line 27 -page 10, line 10 ---	1-8
X	US 5 438 571 A (CROUCH SIMON E ET AL) 1 August 1995 (1995-08-01) abstract column 2, line 26 - line 46 column 11, line 47 -column 13, line 56 column 14, line 35 - line 41 claims --- -/--	1-8



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

### \* Special categories of cited documents :

- \*A\* document defining the general state of the art which is not considered to be of particular relevance
- \*E\* earlier document but published on or after the international filing date
- \*L\* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- \*O\* document referring to an oral disclosure, use, exhibition or other means
- \*P\* document published prior to the international filing date but later than the priority date claimed

\*T\* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

\*X\* document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

\*Y\* document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

\* & \* document member of the same patent family

Date of the actual completion of the international search

19 December 2000

Date of mailing of the international search report

02.01.2001

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl.  
Fax: (+31-70) 340-3016

Authorized officer

Koukourlis, S

# INTERNATIONAL SEARCH REPORT

International Application No  
PCT/US 00/19979

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 280 500 A (MAZZOLA MARIO ET AL) 18 January 1994 (1994-01-18) abstract column 2, line 34 -column 3, line 19 figure 3 ---	1-8
A	US 3 649 915 A (MILDONIAN HARRY ASTOUR JR) 14 March 1972 (1972-03-14) column 1, line 6 - line 62 column 2, line 33 -column 3, line 20 column 5, line 24 - line 44 claims figures 1,7 -----	1-8

# INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US 00/19979

## Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:  
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☒ Claims Nos.: 9-123  
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:  
see FURTHER INFORMATION sheet PCT/ISA/210
3. ☐ Claims Nos.:  
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

## Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☐ No protest accompanied the payment of additional search fees.

## INTERNATIONAL SEARCH REPORT

International Application No. PCT/US 00 19979

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

Continuation of Box I.2

Claims Nos.: 9-123

In view of the large number and also the wording of the claims presently on file, which render it difficult, if not impossible, to determine the matter for which protection is sought, the present application fails to comply with the clarity and conciseness requirements of Article 6 PCT (see also Rule 6.1(a) PCT) to such an extent that a meaningful search is impossible. Consequently, the search has been carried out for those parts of the application which do appear to be clear (and concise), namely a method for communicating information by using a plurality of symbols generated by a source device on a network for transmission in the network by the source device and reception in the network by a destination device, as claimed in claims 1-8.

The applicant's attention is drawn to the fact that claims, or parts of claims, relating to inventions in respect of which no international search report has been established need not be the subject of an international preliminary examination (Rule 66.1(e) PCT). The applicant is advised that the EPO policy when acting as an International Preliminary Examining Authority is normally not to carry out a preliminary examination on matter which has not been searched. This is the case irrespective of whether or not the claims are amended following receipt of the search report or during any Chapter II procedure.

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 00/19979

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9734397 A	18-09-1997	GB 2311441 A EP 0824817 A JP 11506293 T US 6052390 A	24-09-1997 25-02-1998 02-06-1999 18-04-2000
US 5438571 A	01-08-1995	US 5550836 A JP 7321839 A US 5598406 A CA 2101860 A EP 0596523 A EP 0714191 A JP 6216925 A US 5583872 A	27-08-1996 08-12-1995 28-01-1997 07-05-1994 11-05-1994 29-05-1996 05-08-1994 10-12-1996
US 5280500 A	18-01-1994	NONE	
US 3649915 A	14-03-1972	BE 768736 A CA 925212 A DE 2130443 A FR 2099293 A GB 1327496 A SE 415430 B	03-11-1971 24-04-1973 30-12-1971 10-03-1972 22-08-1973 29-09-1980



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**